

43

LE JOURNAL QUI TURBO-FORTHe A COEUR OUVERT

MARS 1988

RAYON X

EDITORIAL

Les logiciels proposés actuellement font penser au marché de l'automobile: avant il y avait le moteur et la carrosserie, maintenant on a rajouté l'auto-radio, l'éclairage indirect de lecteur de carte, la télécommande de serrure... Il en est de même sur les micros: avant il y avait CP/M avec ses 12k, puis MSDOS 2.0 avec 50k, après MSDOS 3.3 avec près de 100k, et maintenant OS/2 qui fait plus de 500k. Cette surenchère sert-elle l'utilisateur ou le fabricant pour justifier le prix d'un système par la taille de sa mémoire? Et cette folie gagne les tableurs, les traitements de texte, les langages.... Qui n'a pas sa version 2.n, 2.n+1..5.0, 5.1, TVA non comprise et incompatibilité partielle garantie avec la précédente version.

Le vrai virus ne viendrait-il pas de cette inflation qui pousse à la disparité entre utilisateurs: dBASE II c'est bien, dBASE III, c'est mieux, dBASE III+ c'est encore mieux, dBASE IV c'est.... (attendez, je cherche dans le Larousse des superlatifs) ...beaucoup plus mieux! Le temps de vous familiariser avec un progiciel et déja vous n'êtes plus à la page. Le virus dans ce foisonnement provient plus fréquemment des bogues non détectées à la conception et dûes à une commercialisation hâtive mais nécessaire dans un marché évoluant aussi rapidement. Tirer le premier pour ne pas être coulé, et tant pis pour l'utilisateur final.

Alors si vous cherchez un langage de conception logicielle performant, convivial, modulaire, modifiable, extensible, structuré, polyvalent.. (je cherche encore des superlatifs) ...bref, génial, il n'en reste qu'un: TURBO-Forth. Citez-moi un autre langage qui permet de vérifier une procédure sans exécuter tout le programme, un langage autorisant la mise au point et la vérification en cours de compilation, qui peut interpréter et compiler le contenu d'un fichier, pouvant s'interfacer à tout y compris ce qui n'existe pas encore et qui, par la foi que mettent ses concepteurs et ses utilisateurs en lui, devient chaque jour plus séduisant.

TURBO-Forth est un merveilleux jouet, mais un jouet intelligent. Tout est dévoilé dans les fichiers source, un atout inexistant avec les autres langages. TURBO-Forth bénéficie d'une assistance télématique interactive. TURBO-Forth voit sa bibliothèque logicielle croître chaque jour. Si nous, les concepteurs de TURBO-Forth, avions été aussi nombreux que ceux de TURBO-Pascal, et disposant des mêmes moyens et connaissances, il n'y aurait peut-être maintenant plus qu'un seul langage de conception: TURBO-Forth. Alors amis du "Futur Numérique Proche", n'attendez pas que TURBO-Forth vous rattrape, adoptez-le de suite et découvrez sa formidable puissance. Choisissez entre la 2CV (DÖSCHWO pour les germaniques) et la FERRARI TURBO-Forth, vous ne le regretterez pas.

SOMMAIRE

Nos coordonnées: ASSOCIATION JEDI, 17, rue de la Lancette 75012 PARIS
tel président: (1) 43.40.96.53
tel secrétaire: (1) 46.56.33.67 (attention, changement prochain de n°)

Téléchargement et Forum FORTH: 3615 SAM*JEDI

FORTH

CRIBLE D' ERATHOSTHENE BENCHMARK

revu par M.ZUPAN

Pour: TURBO-Forth 83-Standard
adaptable: F83 Laxen et Perry MSDOS
VolksFORTH

Téléchargement: 3615 SAM*JEDI

LISTING DU PROGRAMME: ERATHOS.FTH

```
echo off
FORTH DEFINITIONS DECIMAL
8190 CONSTANT SIZE
\ pseudo-constante taille table 1 à n/2

: PREMIERS  ( n -- )
\ affiche nombre de premiers de 1 à n
    4 MAX DUP 2/ 1- IS SIZE PAD SIZE 1 FILL
    2 SIZE 0
    DO   PAD I + C@
         IF    I 2* 3 + DUP I +
               BEGIN  DUP SIZE <
               WHILE  0 OVER PAD + C! OVER +
               REPEAT DROP DROP 1+
         THEN
    LOOP
    .  ." Premiers entre 1 et " . ;

: LISTE ( -- )
\ affiche liste des nombres premiers calculés
\ par PREMIERS
    CR 1 . 2 . SIZE 0
    DO   PAD I + C@
         IF I 2* 3 + . THEN
         STOP? ?LEAVE
    LOOP ;
EOF
```

Ce petit programme est un test bien connu illustrant la rapidité du FORTH :

10000 PREMIERS affiche le nombre de nombres premiers compris entre 1 et 10000 comptés selon le très classique algorithme du crible d'Erathosthène (vers 284-192 av. J.C.)

Vous pouvez contrôler la réalité d'un comptage N PREMIERS en demandant à Forth d'afficher la liste des nombres en tapant LISTE.

FORTH

VARIABLES LOCALES REVUES ET CORRIGEES

auteur: Harvey GLASS
adaptation F83: Marc PETREMANN
modifs vocabulaire: Michel ZUPAN

pour : TURBO FORTH
F83 LAXEN et PERRY CP/M et MSDOS
adaptable: VolksFORTH ATARI

Téléchargement: 3615 SAM*JEDI

Ce programme a déja été diffusé dans JEDI. Mais la première version n'était pas sécurisée. En effet, un nombre trop important de variables locales grignotait le vocabulaire FORTH et plantait le système. M.ZUPAN a apporté une correction tout en conservant la syntaxe des mots utilisés <DECLARE <DEFINE VAR LOCAL TEMP et >>.

En cas d'erreur de compilation, il suffit de taper DECLARE> pour 'recoudre' son vocabulaire sans dégats.

Si vous avez déja exploité dans des programmes les variables locales, vous n'aurez rien à y modifier.

LISTING DU PROGRAMME: ALEPH.FTH

```
ONLY DEFINITIONS ALSO FORTH VOCABULARY ALEPH
ONLY FORTH ALSO ALEPH ALSO DEFINITIONS HEX

CREATE #LST 50 ALLOT    \ pile lambda
       #LST 4E +
CONSTANT #LSO           \ point de débordement
VARIABLE #PTR           \ pointeur pile lambda
VARIABLE DP-SAVE        \ sauvegarde de HERE

: >L   (S n -- )        \ empile lambda
  #PTR DUP @ DUP 2- DUP #LST > IF ROT ! !
  ELSE TRUE ABORT" pile lambda pleine" THEN ;
: L>   (S -- n )        \ dépile lambda
  #PTR DUP @ 2+ DUP ROT ! @ ;
: LS@  (S m -- n )      \ lecture lambda
  #PTR @ + @ ;
: LS!  (S n m -- )      \ stockage lambda
  #PTR @ + ! ;
: #POP (S n -- ) \ suppression de n valeurs sur lambda
  #PTR +! ;

DECIMAL
 4 CONSTANT 4    6 CONSTANT 6    8 CONSTANT 8
10 CONSTANT 10  12 CONSTANT 12 14 CONSTANT 14
: [2-14]   (S n -- )
\ accélère l'accès aux premières locales
  CASE  2 OF ['] 2 ,   ENDOF
        4 OF ['] 4 ,   ENDOF
        6 OF ['] 6 ,   ENDOF
        8 OF ['] 8 ,   ENDOF
       10 OF ['] 10 ,  ENDOF
       12 OF ['] 12 ,  ENDOF
       14 OF ['] 14 ,  ENDOF
       DUP [COMPILE] LITERAL ENDCASE ;

HEX
FORTH DEFINITIONS
: <DECLARE  (S -- ) \ débute une déclaration ALEPH
  #LSO #PTR !
  CURRENT @ DUP ['] ALEPH >BODY = IF CR
  TRUE ABORT" <DECLARE ignoré : ALEPH ne peut recevoir une
définition ALEPH"
  THEN AVOC !
  ALSO ALEPH ALSO DEFINITIONS
  HERE DP-SAVE !
  400 ALLOT \ espace réservé à la définition aleph
  0 >L ;

: DECLARE>  (S -- )
\ rétablit les vocabulaires et oublie les locales
  PREVIOUS PREVIOUS
  AVOC @ DUP CURRENT ! CONTEXT !
  DP-SAVE @ (FORGET) ;

\ en cas d'erreur dans une définition ALEPH il faut
\ exécuter un DECLARE> pour récupérer l'espace utilisé
\ sauf s'il s'agit de l'erreur 'ALEPH ne peut recevoir
\ une définition ALEPH'

: VAR   ( <mot> -- )
\ définition d'une variable ALEPH transitoire
  CREATE L> 1+ DUP >L 2* C, IMMEDIATE
  DOES> C@ [2-14] COMPILE LS@ ;

: LOCAL  ( <mot> -- ) \ définition d'une variable locale
  CREATE L> 1+ 100 + DUP >L OFF AND 2* C, IMMEDIATE
  DOES>  C@ [2-14] COMPILE LS@ ;

: LET   ( valeur variable -- )
\ affectation d'une variable locale
  ['] LS! HERE 2- ! ; IMMEDIATE

: <DEFINE ( <mot> -- )
\ définition utilisant les variables ALEPH
  DP-SAVE @ DP !
  AVOC @ DUP CURRENT ! CONTEXT !
  !CSP
  CREATE HIDE 2 LS@ 100 / ?DUP
```

```
   IF BEGIN COMPILE 0 1- ?DUP 0= UNTIL THEN
   2 LS@ OFF AND ?DUP
   IF BEGIN COMPILE >L 1- ?DUP 0= UNTIL THEN
   ] ['] : @ LAST @ NAME> ! ;

: >>  (S -- ) \ fin de définition sous ALEPH
   2 LS@ OFF AND 2* [2-14] COMPILE #POP
   [COMPILE] ;
   HERE DP-SAVE !
   DECLARE> ; IMMEDIATE

ONLY FORTH ALSO DECIMAL

EOF  \ Mode d'emploi en fin de ce fichier.
```

Le module ALEPH permet d'utiliser des variables locales déclarées dans un esprit "Pascal-like". Il est destiné à faciliter l'écriture de définitions complexes pour lesquelles les manipulations de la pile (dup over swap rot pick ...) seraient délicates. On utilise ici une pile accessoire dite pile lambda dont les valeurs sont déclarées sous forme de variables locales c.a.d à définitions transitoires. Un exemple vous expliquera mieux la syntaxe :

FORTH

DIVISION A PRECISION INFINIE

par Michel ZUPAN

pour : TURBO-Forth
F83 Laxen et Perry MSDOS et CP/M
adaptable: VolksFORTH ATARI

Ce programme est situé après EOF dans ALEPH.FTH lequel est téléchargeable depuis 3615 SAM*JEDI.

DIVISE affiche le résultat d'une division 'décimale' signée dans la base courante avec TOUS les chiffres après la virgule. Mieux que la virgule flottante, non? Rassurez-vous: si la suite est infinie, vous pourrez l'arrêter au clavier.

LISTING DE DIVISE:

```
    <declare var    dividende
             var    diviseur
             local  reste

    <define  divise  (S n1 n2 -- )
             diviseur  0= abort" zéro pour vous !"
             cr dividende  .  ." divisé par "
                diviseur   .  ." = "
             dividende 0< diviseur 0< xor
                if ascii - emit then
             diviseur  abs diviseur let
             dividende abs diviseur /mod
             (.) type ascii . emit
             reste let
             begin
                reste stop? not and while
                reste base @ *
                diviseur /mod (.) type
                reste let
             repeat
             >>
```

Si vous ré-écrivez ce DIVISE en FORTH usuel, comptez les opérateurs de pile (dup rot etc...): vous comprendrez à quoi peut servir le vocable ALEPH.

FORTH

PRODUIT 32 BITS NON SIGNE

par Marc PETREMANN

Pour : TURBO-Forth
F83 LAXEN et PERRY MSDOS et CP/M
Adaptable : VolksFORTH ATARI
complément: 79-Standard (en fin de programme)
non diffusé en téléchargement.

LISTING DE XD*

```
\ soit à exécuter  ud3 := ud1 * ud2 , on pose
\                ud1h   ud1l  h pour high, l pour low
\                ud2h   ud2l
\ -------------------------
\                t1h    t1l     t1:=ud1l*ud2l   en 32 bits
\         t2h    t2l     0      t2:=ud1h*ud2l   en 32 bits
\         t3h    t3l     0      t3:=ud1l*ud2h   en 32 bits
\ t4h     t4l     0             t4:=ud1h*ud2h   en 32 bits
\ -------------============     ne sera exploité que:
\ inexploité    ud3h   ud3l    ud3:=T1+(t2l*256)+(t3l*256)

<DECLARE   VAR ud1l   VAR ud1h   VAR ud2l   VAR ud2h
           LOCAL t1l  LOCAL t1h  LOCAL t2l  LOCAL t2h
           LOCAL t3l  LOCAL t3h  \ LOCAL t4l  LOCAL t4h
                                   ( inexploitées)
<DEFINE   XD*  ( ud1 ud2 --- ud3 )
\ produit de deux ud non signés
            ud1l ud2l UM*  t1h LET  t1l LET
            ud1h ud2l UM*  t2h LET  t2l LET
            ud1l ud2h UM*  t3h LET  t3l LET
         \  ud1h ud2h UM*  t4h LET  t4l LET
            ( résultat inexploité)
 \ somme des produits partiels
            t1l t1h t2l 0 D+  t3l 0 D+  >>

\ plus compréhensible que la définition suivante:
\ : XD*  ( ud1 ud2 --- ud3)   ( en FORTH 83-Standard)
\    OVER  4 PICK  UM* >R >R
\    3 ROLL UM* DROP  >R  UM* DROP
\    0 SWAP R>  0 SWAP D+  R> R>  D+  ;
\ : XD*  ( ud1 ud2 --- ud3)   ( en FORTH 79-Standard)
\    OVER  5 PICK  U* >R >R
\    4 ROLL U* DROP  >R  U* DROP
\    0 SWAP R>  0 SWAP D+  R> R>  D+  ;
\ Exemple d'exécution:
\  35000. 20000. XD* UD.  affiche 700000000
```

FORTH

UTILITAIRES DE GESTION DE FICHIERS

par Michel ZUPAN

Pour : TURBO-Forth 83 Standard
Adaptable: partiellement F83-Laxen et Perry MSDOS

Téléchargement: 3615 SAM*JEDI

Ce programme remplit diverses fonctions:

- savoir si un fichier est autorisé en lecture ou écriture.
- modifier ce type d'accès quand survient l'erreur 'accès interdit'.
- récupérer et modifier les date et heure de mise à jour d'un fichier.
- pouvoir tester la taille d'un fichier existant.
- utiliser des fichiers en accès direct: ouvrir un tel fichier, lire ou écrire des enregistrements, allonger le fichier, le fermer.
- lire et compiler les fichiers blocs du F83 de Laxen et Perry.
- convertir un fichier blocs F83 en fichier texte TURBO-F83.

REMARQUE IMPORTANTE: ce programme est disponible dans le module M3 du package TURBO-Forth en compagnie d'autres programmes déjà diffusés dans JEDI. Le prix du module M3 est de 37,00 Fr.

LISTING DU PROGRAMME: UFILES.FTH

```
\ ***********************************************************
\ 1) Utilitaires d'attributs de fichiers
```

```
\ ***********************************************************
HEX  \ primitives ms-dos :
CODE (USING) ( att adr-pathname -- fl)
\ modification d'attribut
     DX POP   CX POP    4301 # AX MOV   21 INT
     U>= IF   0 # AX MOV   THEN 1PUSH
     END-CODE

CODE (USING?) ( adr-pathname -- att fl )
\ interrogation d'attribut
     DX POP   4300 # AX MOV   21 INT   CX PUSH
     U>= IF   0 # AX MOV   THEN 1PUSH
     END-CODE

DECIMAL  \ mots utilisateur :
: USING ( <filename> att -- )
  \ détermine l'attribut d'un fichier
     FILENAME (USING) ?DOS-ERR ;

: USING? ( <filename> -- att )
  \ délivre l'attribut d'un fichier
     FILENAME (USING?) ?DOS-ERR ;

\ L'attribut d'un fichier est un octet dont les
\ bits 0 à 5 sont à 1 si:
\  bit 0 : fichier utilisable en lecture seule
\  bit 1 : fichier caché (non affichable, non effaçable)
\  bit 2 : fichier système ( recopiés par SYS )
\  bit 3 : fichier nom du volume
\  bit 4 : répertoire
\  bit 5 : bit d'archivage indiquant que le fichier a
\          été modifié
\ ces attributs sont en partie panachables; certaines
\ modifications d'attributs sont interdites (ex: un
\ répertoire ne peut être changé en simple fichier).

\ ***********************************************************
\ 2) Utilitaires de date et heure d'un fichier
\ ***********************************************************

HEX  \ primitives ms-dos :
CODE (FILEDATE)   ( dtime hndl -- fl )
\ modification date et heure
     BX POP   CX POP   DX POP   5701 # AX MOV   21 INT
     U>= IF   0 # AX MOV   THEN 1PUSH
     END-CODE

CODE (FILEDATE?)  ( hndl -- dtime fl )
\ interrogation date et heure
     BX POP   5700 # AX MOV   21 INT   DX PUSH   CX PUSH
     U>= IF   0 # AX MOV   THEN 1PUSH
     END-CODE

DECIMAL
: DT>FDT  ( jour mois année heure min -- dtime )
     32  * SWAP 2048 * + >R
     1980 - 0 MAX 512 * SWAP 32 * + + R> ;

: .FDT    ( dtime -- )
     BASE @ >R DECIMAL
     SWAP 0 512 UM/MOD 1980 + SWAP 32 /MOD SWAP
     (.) TYPE ASCII - EMIT (.) TYPE ASCII - EMIT .
     0 2048 UM/MOD 0 <# # # #> TYPE ASCII : EMIT
     32 / 0 <# # # #> TYPE SPACE R> BASE ! ;

\ mots utilisateurs:
: FILEDATE ( <filename> jour mois année heure min -- )
     5 ?ENOUGH DT>FDT OPEN DUP >R (FILEDATE)
     ?DOS-ERR R> (CLOSE) ;

: FILEDATE? ( <filename> -- )
\ affiche date et heure fichier
     OPEN DUP (FILEDATE?) ?DOS-ERR .FDT (CLOSE) ;

\ ***********************************************************
\ 3) détermination de la longueur d'un fichier
\ ***********************************************************
: FILESIZE ( <filename> -- dlen )
     OPEN DUP 0 0 ROT 2 (SEEK) ?DOS-ERR ROT (CLOSE) ;

\ Exemple
\ Soit un fichier ESSAI.TXT dans le répertoire courant
\
\ 1 USING ESSAI.TXT
\        autorise ESSAI.TXT en lecture et écriture
\ USING?  ESSAI.TXT .
\        affiche 1 l'attribut donné à ESSAI.TXT
\ 1 1 1988 15 30 FILEDATE ESSAI.TXT
\        date au 1/1/88 15h30 le fichier ESSAI.TXT
\ FILEDATE? ESSAI.TXT
\        affiche alors 1-1-1988 15:30
\ FILESIZE ESSAI.TXT UD.
\        affiche la longueur du fichier ESSAI.TXT

\ ***********************************************************
\ 4) Gestion de fichiers en accès direct
\ ***********************************************************
VARIABLE <FILE
\ ticket du fichier utilisé en accès direct
VARIABLE B/REC      \ longueur d'un enregistrement
VARIABLE CAPACITY
\ nombre d'enregistrements dans le fichier
-1024 LBUF FILES 6 + * -
CONSTANT LIMIT
\ fin du tampon de lecture-écriture des accès directs
\ placé en dessous des tampons réservés à TURBO-FORTH
: ?FILE-R  ( -- )
\ vérifie ouverture d'un fichier accès direct
   <FILE @ 0=
   ABORT" pas de fichier ouvert en accès direct" ;
: LOCATE  ( n -- )
\ positionne le pointeur fichier à l'enregistrement n
   ?FILE-R DUP CAPACITY @ U>
   ABORT" enregistrement hors limite"
   B/REC @ UM* <FILE @ 0 (SEEK) ?DOS-ERR 2DROP ;

\ mots utilisateur :
\ ouverture, lecture, écriture, allongement, fermeture

: OPEN-R ( <filename.ext> b/rec -- )
\ ouvre ou crée un fichier en accès direct
   B/REC !  \ précisant la longueur d'enregistrement
   ?OPEN  PATHWAY
   DUP 0 (SEARCH0) IF 2 (OPEN)
                   ELSE 0 (CREATE)
                   THEN ?DOS-ERR
   DUP 0 0 ROT 2 (SEEK) ?DOS-ERR
   B/REC @ UM/MOD CAPACITY ! DROP
   <FILE ! ;

: RECORD   ( -- adr long)
\ tampon lecture-écriture d'enregistrement
   B/REC @ LIMIT OVER - SWAP ;

: READ ( n -- )
\ lit le n-ième enregistrement depuis le fichier
   LOCATE DSEGMENT RECORD <FILE @ (GET) ?DOS-ERR DROP ;

: WRITE ( n -- )
\ écrit le n-ième enregistrement dans le fichier
   LOCATE DSEGMENT RECORD <FILE @ (PUT) ?DOS-ERR DROP ;

: APPEND ( -- ) \ ajoute le tampon en fin de fichier
   ?FILE-R 0 0 <FILE @ 2 (SEEK) ?DOS-ERR 2DROP
   DSEGMENT RECORD <FILE @ (PUT) ?DOS-ERR
   B/REC @  <> ABORT" fichier trop grand"
   CAPACITY 1+! ;

: CLOSE-R ( -- ) \ ferme le fichier en accès direct
   ?FILE-R <FILE DUP @ (CLOSE) OFF ;

\ couche ultérieure:
\   définition de champs dans un enregistrement
: FIELD ( <nom-de-champ> position longueur -- )
   CREATE , ,
   DOES> ( -- adr l ) RECORD DROP OVER 2+ @ + SWAP @ ;

\ exemple:
\ (sans vouloir développer ici une base de donnée...)
\ : ANNUAIRE " 100 OPEN-R ANNUAIRE.DAT" $EXECUTE ;
\  0 20 FIELD NOM
\ 20 70 FIELD ADRESSE
\ 90 10 FIELD TELEPHONE
\ : INSCRIT ANNUAIRE RECORD BLANK
\           CR ." nom     : "  NOM      EXPECT
\           CR ." adresse : "  ADRESSE  EXPECT
```

```
\           CR ." tél.    : " TELEPHONE EXPECT
\           APPEND CLOSE-R ;
\ : .TEL    ANNUAIRE CAPACITY @ 0 DO I READ
\           CR NOM TYPE TELEPHONE TYPE LOOP CLOSE-R ;
\ etc..

\ ***********************************************************
\ 5) Interprétation/compilation de fichiers-blocks
\    du F83 Laxen et Perry
\ ***********************************************************
VARIABLE BLK      \ bloc en cours de compilation
VARIABLE LOADING?
\ booléen pour empêcher ré-entrance de LOAD

: OPEN-BLK  ( <filename.ext> -- )
\ ouvre un fichier-blk F83-LP
  1024 OPEN-R BLK ON ;

: BLOCK ( n -- adr ) \ adresse du bloc n
  DUP BLK @ = IF DROP ELSE DUP READ BLK ! THEN  RECORD DROP
;

: -->  ( -- ) \ passe à l'écran F83-LP suivant
  BLK @ 1+ BLOCK DROP >IN OFF ; IMMEDIATE

warning off
: \ ( -- )
\ nouvelle définition valable dans les deux
\ modes LP et TURBO
    >IN @ NEGATE LOADING? @
    IF 64 ELSE C/L THEN MOD >IN +! ; IMMEDIATE
warning on

: LOAD ( <filename.ext> n -- )
\ interprète le n-ième écran du fichier F83-LP
  LOADING? @ HANDLE @ 0< AND ABORT" LOAD non ré-entrant"
  OPEN-BLK LOADING? ON BLOCK 1024 $EXECUTE
  LOADING? OFF CLOSE-R ;

\ notez la syntaxe particulière de ce LOAD qui n'étant
\ pas ré-entrant mixe les anciens mots OPEN et LOAD:
\   1 LOAD PROG.BLK  compile l'écran 1 du fichier
\ PROG.BLK du F83-LP et éventuellement les suivants si
\ s'y trouvent --> .
\ Le même type de syntaxe est adopté pour les mots
\ permettant de visualiser le contenu de ce type de
\ fichiers-blk:

: INDEX ( <filename.ext> -- )
\ Affiche les index (lignes 0) d'un fichier LP
    OPEN-BLK CAPACITY @ 0
    DO CR I 3 .R SPACE I BLOCK 64 -TRAILING TYPE
    STOP? ?LEAVE LOOP CLOSE-R ;

: LISTING ( <filename.ext> n1 n2 -- )
\ liste écrans n1 à n2 d'un fichier LP
    2 ?ENOUGH OPEN-BLK 1+ SWAP
    DO I BLOCK CR ." scr# " I . FALSE
       16 0 DO DROP CR I 64 * OVER + 64 -TRAILING TYPE
            STOP? IF TRUE LEAVE ELSE FALSE THEN LOOP NIP
       ?LEAVE CR LOOP CLOSE-R ;

\ ***********************************************************
\ 6) Conversion de fichiers .BLK  F83-LP
\  en fichiers texte
\ ***********************************************************
\ UNBLOCK fichier-blk-source  fichier-texte-cible

: UNBLOCK ( <filename1.ext> <filename2[.ext]> -- )
  OPEN-BLK ?OPEN FILENAME 0 (CREATE) ?DOS-ERR
  CAPACITY @ 0
  DO I BLOCK  16 0
     DO DUP I 64 * + 64 -TRAILING ?DUP
        IF PAD PLACE
           [ HEX ] 0A0D
           [ DECIMAL ] PAD COUNT + !  ( + crlf )
           PAD COUNT 2+ TUCK DSEGMENT -ROT 5 PICK (PUT)
           ?DOS-ERR <> ABORT" fichier trop long"
        ELSE DROP
        THEN
     LOOP DROP
  LOOP (CLOSE) CLOSE-R ;
```

```
\ Ex :  UNBLOCK  B:GENIUS.BLK  A:\FORTH\GENIUS
\        crée un nouveau fichier texte
\ A:\FORTH\GENIUS.FTH contenant toutes les
\ lignes non vides du fichier-blk
\ B:GENIUS.BLK . Ce nouveau fichier est INCLUDable
\ par TURBO après éventuelle édition (suppression des
\ THRU et --> notamment ).
EOF \ FIN DU PROGRAMME
```

FORTH

RENDRE TURBO FORTH RESIDENT

par Michel ZUPAN

Pour : TURBO-Forth 83 Standard
Adaptable: partiellement F83 Laxen et Perry MSDOS

But: sortie de Turbo-Forth maintenu RESIDENT en mémoire, retour par interruption logicielle.

REMARQUE IMPORTANTE: ce programme est disponible dans le module M3 du package TURBO-Forth et en téléchargement sur 3615 SAM*JEDI.

LISTING DU PROGRAMME: HALT.FTH

```
ONLY FORTH ALSO HIDDEN ALSO DEFINITIONS HEX
\ primitive de sortie avec programme laissé résident
CODE (HALT) ( taille-résidente-en-paragraphes  -- )
     DX POP   3100 # AX MOV   21 INT   NEXT END-CODE

\ modification de ALLOC pour n'importe quel PSP
LABEL RES-ALLOC ( taille -- dernier_seg )
     62 # AH MOV   21 INT   BX ES MOV
     \ récupère PSP courant
     BX POP   4A # AH MOV   21 INT   1PUSH
RES-ALLOC ' ALLOC !

50 CONSTANT RES-INT
\ n° de l'interruption de "retour en forth"

FORTH DEFINITIONS

\ sortie de Forth laissé résident avec mot de relance
: HALT ( <mot-de-relance> -- )
  BL WORD COUNT 80 PLACE
  \ un mot ou rien dans le PSP de Turbo
  0 RES-INT 4 * 2 + L@ DSEGMENT <>
  \ vecteur RES-INT déjà affecté ?
  IF 100 0 RES-INT 4 * L!
     \ non : initialise vecteur sur Forth
     DSEGMENT 0 RES-INT 4 * 2 + L!
     HEIGHT DUP ALLOC DROP
     \ limite taille minimale du forth
     (HALT)  \ et sortie résidente
  ELSE BYE   \ oui : simple BYE
  THEN ;
CR
.( commande HALT [<mot>] active pour sortie résidente )

\ création d'un micro-fichier .COM (19 octets)
\ de relance du Forth

:: filename 10 (search0)
   if only forth also decimal eof then ; RESUME.COM
\ cette commande interprètée stoppe ici
\ la compilation de ce fichier si RESUME.COM
\ existe déjà dans le répertoire courant.

CR .( création d'un fichier RESUME.COM )
CR .( pour revenir au Forth )
warning off
LABEL RESUME
\ code du micro-programme de relance du forth
  0 # AX MOV   AX DS MOV   142 #) AX MOV
  AX AX OR    0<>
  IF RES-INT INT
     \ exécute une interruption 50H
  THEN 4C00 # AX MOV 21 INT
  \ fin de fichier si pas de INT50
```

```
RESUME HERE SAVE RESUME.COM
FORGET RESUME
ONLY FORTH ALSO DEFINITIONS DECIMAL

echo on warning on eof

EOF \ FIN DU PROGRAMME
```

Ce fichier est un exemple de la technique consistant à laisser un programme résident en mémoire, en l'occurence tout TURBO-FORTH ou l'une de ses applications. HALT est une espèce de BYE qui quitte TURBO avec retour au programme appelant (COMMAND.COM du DOS le plus souvent) mais laisse TURBO résident en mémoire.

HALT peut être en outre suivi d'un mot qui sera exécuté dès la relance du Forth laissé en mémoire. Exemple HALT WORDS, HALT WARM ou simplement HALT <cr> pour une relance boot.

Le problème avec un programme résident est la façon de le relancer. Le procédé retenu ici est le plus simple: TURBO sera relancé par une interruption logicielle. HALT initialise le vecteur de l'interruption 50h qui n'est pas utilisée en pratique courante (un programme externe peut éventuellement déjà l'utiliser: vérifiez alors que les adresses 0:140 à 0:143 de la table des interruptions sont à zéro. Au besoin changez de numéro d'interruption).

TURBO-FORTH peut être réactivé par n'importe quel programme lançant une INT50.

Le petit fichier RESUME.COM exécute une simple interruption 50h. C'est plus rapide que de recharger TURBO et de compiler à nouveau une application. Si le vecteur 50h n'a pas été initialisé par un HALT, le lancement de RESUME ne provoque rien.

Quand on utilise un programme aussi complexe que TURBO-FORTH en mode résident il devient parfois difficile de s'y retrouver dans les segments et les zones mémoires allouées! Voici quelques explications:

- une fois le Forth relancé par la commande RESUME, sachez que votre programme en cours est RESUME.COM et non plus TURBO.COM. Ceci est important car le PSP (octets 0 à 100h Préfixe Segment de Programme) n'est plus dans le segment du Forth. Il peut en découler certaines modifications notamment si on utilise l'intégrateur PROGRAM ou si surviennent des erreurs dans des fichiers ouverts (le mot ALLCLOSE ferme tous les fichiers de Forth, pas ceux d'un autre programme: utilisez la primitive handle (CLOSE) pour fermer des fichiers restés ouverts.)

- ce changement de PSP explique le "patch" d'une nouvelle définition de ALLOC pour permettre à RESUME d'utiliser lui-aussi l'intégrateur TURBO. En effet la définition de ALLOC dans TURBO prend le segment courant comme segment du PSP. Cette nouvelle définition, qui utilise une fonction MSDOS disponible seulement à partir des versions 3.x, récupère le segment PSP du programme en cours. Vous n'en aurez pas besoin si vous ne devez pas utiliser l'intégrateur dans votre application. Dans ce cas les commandes d'intégration fourniront un message d'erreur "mémoire insuffisante".

- une deuxième utilisation de HALT dans RESUME rendrait celui-ci résident ce qui bien-sûr n'a guère d'intérêt et doublerait l'espace réservé. Aussi dans ce cas, c'est un classique BYE qui est exécuté (Forth reste évidemment toujours résident et "resumable"). Un deuxième HALT servira seulement à modifier le mot de relance.

On peut bien-sûr imaginer obtenir une relance Forth plus ergonomique comme de placer un autre résident qui testerait l'appui d'une touche, sauverait tout l'environnement comme l'écran, le curseur et le programme en cours avant de relancer TURBO pour revenir par une fin d'interruption...

Ce test du clavier par détournement d'une interruption existante telle celle de l'horloge pourrait d'ailleurs se trouver dans l'application TURBO elle-même...

COMMUNICATION

LA LIAISON MINITEL - PC

par Marc PETREMANN

Si vous faites partie de ces veinards qui ont été équipés gracieusement par FRANCE TELECOM d'un minitel et qui ont la chance de disposer également d'un compatible PC, voici une rubrique très alléchante.

Bien sûr, il n'est pas question de parler des messageries roses, mais plutôt de l'aspect utilitaire de la communication télématique et de l'environnement informatique nécessaire pour exploiter les données reçues.

UNE PREMIERE SOLUTION: LA CARTE TELEMATIQUE

De nombreux constructeurs proposent des cartes d'extensions permettant de se connecter sur le réseau télématique. Les plus coûteuses travaillent en 300/300 bds, 1200/75 et 75/1200 bds (possibilité de serveur monovoie), 1200/1200 bds, et pour le haut de gamme également en 2400/2400 bds synchrone.

Pour avoir accès aux seuls services télématique, la carte de base (KORTEX, WINTEL, etc...) est à environ 900 Fr accompagnée d'un logiciel de gestion de communication.

La plupart des cartes de base sont équipées de la numérotation automatique (impulsion décimale et fréquence vocale), détection de sonnerie et fonctionnent en 1200/75, 75/1200 et accessoirement en 1200/1200 half ou full duplex. Les cartes mixtes 300/300 et 75/1200.. 1200/1200 sont bien plus chères.

UNE SECONDE SOLUTION: LE CABLE DE LIAISON

Des schémas ont été publiés dans à peu près toutes les revues et des produits sont disponibles à des prix très variables. L'ensemble fonctionnel (câble+logiciel) est disponible généralement à partir de 250 Fr.

Le câble relie la sortie péri-informatique du Minitel à la prise RS232C du compatible PC.

Mais ce qui fait la valeur du produit, dans les deux cas, est la qualité du logiciel proposé. Pour notre part, nous avons essayé le câble LCE-COM entre un minitel (très vieux modèle, même pas retournable...) et un PERSONNA 1600 de LOGABAX (compatible PC/XT).

LE LOGICIEL

Le logiciel gérant le câble LCE-COM est lancé en tapant MINITEL. Un premier programme initialise la liaison RS232C. Ensuite, un menu propose diverses options.

Pour un premier essai, nous avons essayé les cours de la Bourse diffusés par la COTE DESFOSSES:

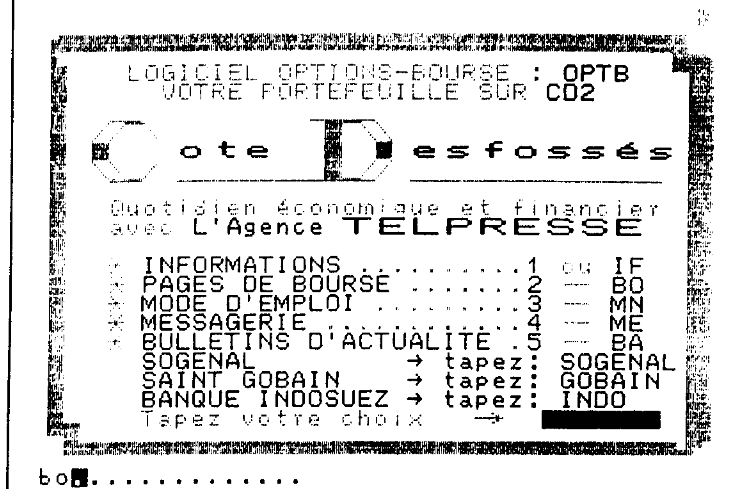
3616 CD

La prise de ligne peut être faite dès réception de la porteuse par appui sur CNX/FIN sur le MINITEL ou par appui de la touche F4. Si votre MINITEL est de type M10, le numéro peut être composé depuis le compatible.

La première procédure expérimentée, la plus intéressante pour notre portefeuille, est la procédure CAPTURE de pages. Dès qu'elle est active, toute transaction est enregistrée en continu dans le fichier ouvert à cet effet. Notre fichier de capture de la COTE DESFOSSES a été nommé pour la circonstance CDEFOS.VTX.

A partir de la prise en ligne, l'affichage du MINITEL apparaît en écho sur l'écran du compatible. Les divers tons de gris sont restitués assez fidèlement, à condition de disposer d'une carte CGA (ce qui est le cas des AMSTRAD

PC 1512). Voici un exemple de l'impression de la page affichée sur l'écran du compatible de présentation du service boursier de la COTE DESFOSSES:



bo...............

Une autre option très intéressante permet la conversion en fichier ASCII d'une page VIDEOTEX. Ainsi, la page ci-dessus devient après conversion:

```
LOGICIEL OPTIONS-BOURSE : OPTB
VOTRE PORTEFEUILLE SUR CD2

// \            \\
     o t e          e s f o s s é s
\\ /            //

Quotidien économique et financier
avec L'Agence T E L P R E S S E

* INFORMATIONS ..........1 ou IF
* PAGES DE BOURSE .......2 -  BO
* MODE D'EMPLOI .........3 -  MN
* MESSAGERIE ............4 -  ME
* BULLETINS D'ACTUALITE .5 -  BA
  SOGENAL           tapez: SOGENAL
  SAINT GOBAIN      tapez: GOBAIN
  BANQUE INDOSUEZ   tapez: INDO
  Tapez votre choix        envoi
```

Ce qui, par parenthèse, nous évite de taper 20 lignes nous même pour le présent article (non, mais...).

Ce texte peut être consulté hors connexion. Pour aller plus loin dans notre exemple, nous avons poursuivi la consultation en choisissant l'option 2 - BO

```
     \          NOUVEAU DANS VOS PAGES
/ /  \/  \\          DE COURS

               * UNE INFORMATION CONCER-
  \  /\  /       NANT CETTE SOCIETE EXISTE
\     /  /          DANS NS OU NR.

   J o u r n é e
       d u   4   M a r s

 * Bourses francaises ......1 ou BF
   Début de mise à jour 10h15

 * Matif....................2 -  MATIF
   Mise à jour 16h00.

 * Bourses étrangères.......3 -  BE
   Mise à jour 9h30, 11h30, 13h15.

 * Marché options négociables 4 - MONA

tapez votre commande.........    ENVOI
bf.............
```

Nous avons ensuite sélectionné l'option BF. On vous épargnera l'affichage des autres pages de présentation. La sélection suivante est MC (Marché Continu). C'est à partir de ce point que les véritables informations apparaissent. Il y a onze pages. La capture en continu de ces pages a pris moins de deux minutes. Puis nous nous sommes déconnectés. Maintenant, consultons nos valeurs:

```
C ote D esfossés                          1/11

   MC aussi en tapant les 3ères lettres


              Marché Continu
           (de 10h00 à 17h00)
           Cours du 04/03/88


Premier  Dernier                     Pré-
          connu        VALEURS       cédent

 235      238      ALSPI.........   238
 248      245,50   ALSTHOM.......   248,80
 827      842      AUXIL.D'ENTR..   834
 632      646      AVIONS M.DAUS.   641
 291      295      BAIL EQUIPment   300
 889      875      BAIL INVESTIS.   885
 430,20   428      BANCAIRE CIE..   428
 300      300      B.H.V.........   301

 T apez un mot clé       Envoi ou suite .
.................
```

Ladite page se présente en réalité comme ceci sur votre écran MINITEL:

Cote Desfossés — 4/11

Marché Continu

Cours du 04/03/88

Premier	Dernier connu	VALEURS	Pré-cédent
480	483,90	D.M.C.........	487
1550	1580	DOCKS FRANCE..	1630
539	536	DUMEZ.........	555
1059	1065	EAUX (Gle des)	1064
960	970	ECCO..........	886
680	676	ELECTRO-FINAN.	682
259,50	259,50	ELF-AQUITAINE.	259,40
2306	2339	ESSILOR INTER.	2340
1485	1510	ESSILOR IN.ADP	1470
290	282	ESSO..........	285
1280	1260	EURAFRANCE....	1275
861	860	EUROCOM S.A...	861
2240	2215	EUROMARCHE....	2241

Tapez un mot clé → [illegible] ou [illegible] .
....

Si vous n'avez que peu de valeurs boursières à contrôler, le pointage sera rapide. Mais le problème est tout autre si vous faites des statistiques au jour le jour. Un fichier ASCII c'est bien joli, mais c'est peu aisé à exploiter, surtout s'il faut tout retaper sur un programme de gestion de base de données.

Ce point délicat est le sujet de l'article qui suit.

dBASE III

CONVERSION DE FICHIERS ASCII EN FICHIERS dBASE III

par Marc PETREMANN

C'est parce que j'ai eu des problèmes de ce genre à résoudre professionnellement que je me suis décidé à en faire un article. Il ne s'agissait pas, dans mon cas, des cours de la bourse, mais des codes nationaux des commutateurs téléphoniques à relever pour une application professionnelle. La quantité d'information à relever était si importante que des erreurs se seraient produites si j'avais dû recopier les codes à la main.

Pour JEDI, j'illustrerai cette technique en reprenant les cours de la bourse tels qu'ils ont été relevés, ceci étant expliqué dans le précédent article.

REMISE EN FORME DES DONNEES EN ASCII

La première opération consiste à remettre en forme le contenu de notre fichier ASCII:

```
                Marché Continu
               Cours du 04/03/88

Premier  Dernier                     Pré-
          connu      VALEURS         cédent

  ...      ...     TELEMEC.ELECT.  5505
 161,50    162     THOMSON-CSF...   165
 342,20    346     TOTAL.........   345,50
  71,70     71,60  TOTAL certif..    71,75
1049      1079     T.R.T.........  1070
 405       405     U.F.B.........   406
 799       784     U.I.C.........   800
 535       532     U.I.F.........   555
 500       500     U.I.S.........   502
 720       720     UNIBAIL.......   730
 182       185     U.C.B.........   185,10
 314,50    295     VIA BANQUE....   313

 T apez un mot clé     Envoi ou * RETOUR
```

(lignes à supprimer : l'en-tête et la dernière ligne)

Il faut rendre jontives toutes les lignes contenant les cotes en supprimant la garniture. Cette opération est réalisée à l'aide d'un traitement de texte (WORDSTAR en mode N -non document- ou WORD ou WORDPERFECT) quelconque. Ce qui donne le fichier résultant:

```
 235       238      ALSPI.........  238
 248       245,50   ALSTHOM.......  248,80
 827       842      AUXIL.D'ENTR..  834
 632       646      AVIONS M.DAUS.  641
--------------------- etc....
 500       500      U.I.S.........  502
 720       720      UNIBAIL.......  730
 182       185      U.C.B.........  185,10
 314,50    295      VIA BANQUE....  313
```

Soit environ 120 lignes de données.

Maintenant, isolons une ligne pour comptage du nombre de caractères des différents champs de la structure à définir sous dBASE III:

```
 248       245,50  ALSTHOM.......  248,80
 314,50    295     VIA BANQUE....  313
-------   -------  --------------  -------
champ 1   champ 2     champ 3      champ 4
```

Taille:

champ 1: 7 caractères
champ 2: 7 caractères
champ 3: 14 caractères
champ 4: 7 caractères

Mais il y a les blancs entre champs. Ils font partie de l'enregistement. Soit on reformate les lignes. C'est possible et même facile avec un traitement de texte comme WORDPERFECT en définissant une macro:

```
 314,50     295     VIA BANQUE....  313
```

devient après suppression des blancs parasites:

```
 314,50 295    VIA BANQUE.... 313
xxxx,xx                              champ1
        xxxx,xx                      champ2
               xxxxxxxxxxxxxx        champ3
                             xxxx,xx champ4
```

Ou bien on définira un sous-fichier en dBASE prenant en compte les espaces parasites:

```
STRUCTURE

  OUVERTURE N  7  2
  BLANC1    C  3
  CLOTURE   N  7  2
  BLANC2    C  1
  VALEUR    C 14
  BLANC3    C  1
  VEILLE    N  7  2
```

soit en définissant un fichier transitoire sans prendre en compte les espaces parasites:

```
  OUVERTURE N  7  2
  CLOTURE   N  7  2
  VALEUR    C 14
  VEILLE    N  7  2
```

Dans les deux cas, nommons notre fichier CDEFOS.DBF. Puis pour charger le contenu du fichier ASCII dans notre fichier dBASE III, il suffit de taper:

```
  APPEND FROM <file.ext> TYPE SDF
```

Le type SDF (pour System Data Format) précise à dBASE que vous reprenez des données provenant d'un fichier ASCII, ici <file.ext> étant remplacé par le nom du fichier ASCII remanié par le traitement de texte. Une fois vos données chargées, vérifiez par un petit coup de LIST que tout s'est bien passé.

Mais vous êtes un passionné de bourse, et vous souhaitez conserver toutes les valeurs au jour le jour. Cette conservation ne peut se faire que dans un fichier contenant l'historique des évolutions des valeurs boursières. La structure de ce fichier doit prendre en compte d'autres données dont la date:

```
STRUCTURE DU FICHIER: BOURSE.DBF
  DATE      D  8
  OUVERTURE N  7  2
  CLOTURE   N  7  2
  VALEUR    C 14
  VEILLE    N  7  2
```

Que vous ayez une structure pour CDEFOS.DBF prenant en compte ou non les blancs parasites, vous utilisez maintenant BOURSE.DBF:

```
  USE BOURSE.DBF
  APPEND FROM CDEFOS.DBF
```

puis initialisez la date vide par:

```
  STORE '04/03/88' TO JOUR
  REPLACE DATE WITH CTOD(JOUR) FOR DTOC()='        '
  *  teste si date contient 8 espaces
```

Et voilà, votre base BOURSE.DBF con

Nota: sur de longues séries de valeurs contigues, le traitement en transformée de Fourier peut être appliqué pour détecter des périodicité dans la fluctuation des valeurs. En temps normal, ces périodicité peuvent être masquées par le "bruit de fond". Il serait intéressant

qu'un adhérent motivé nous fasse un papier sur ce sujet qui ne manquera pas de nous passionner et, qui sait, nous rendre riche (5 oeufs frais au kilo).

COURRIER:
Je quitte un instant mon micro pour vous remercier de m'avoir fait parvenir rapidement de manuel FORTH-83. J'imagine que vous êtes quelque peu débordés par les demandes de disquettes TURBO-Forth et la préparation de la version développeur. Si je puis vous être de quelqu'utilité, je pense à la rédaction d'un résumé thématique des commandes qui me semble un complément indispensable au répertoire alphabétique traditionnel, n'hésitez pas à me le demander.... Le premier contact avec TURBO-Forth est enthousiasmant. Super le positionnement du curseur sur l'erreur au rappel de l'éditeur lors d'une compilation. Barvo les appels de programmes externes quelqu'en soit l'origine. Et l'appel d'une application FORTH avec passage de paramètre, ou l'exécution d'un fichier de commandes FORTH. Tiens, un truc surprenant: depuis le DOS, la commande

```
TURBO INCLUDE TEST.FTH
```

avec en fin de TEST.FTH le mot COLD, boucle plusieurs fois sur la compilation (faites le test avec seulement COLD dans le fichier). Est-ce à dire que la compilation s'est ajoutée à la définition de COLD?! Mais le plus agréable est d'être débarrassé du passage majuscule/minuscules (FIG 79 APPLE II, LMI 83).

Alain LAMBERT - 78140 VELIZY

REPONSE: Le comportement de COLD est tout à fait normal. Lors d'un passage de paramètre depuis DOS, exemple TURBO WORDS, l'exécution de COLD depuis le clavier ou depuis un fichier, relance le mot BOOT, lequel est vectorisé sur HELLO. Or HELLO contient dans les premières lignes, le test de l'octet 128. A cette adresse figure la longueur de la chaîne passée en paramètre. Si cette longueur est non nulle, HELLO réexécute la chaîne à chaque BOOT, HELLO ou COLD.

Concernant la documentation, TURBO-Forth dispose maintenant de son auto-documentation, ce qui peut faciliter la vie des débutants se mettant au FORTH.

LE SECRETAIRE

COURRIER:
(concernant TURBO-Forth)... je suis très intéressé par votre version d'évaluation de TURBO-Forth. C'est une très bonne idée. Cependant, je rejoins un peu Mr JACCOMARD (JEDI n°40 p12) lorsqu'il écrit qu'il est dommage d'écrire son programme avec un traitement de texte externe à FORTH; mon idée serait plutôt d'intégrer un "vrai" traitement de texte sous FORTH, à quand WORDSTAR en Forth?

Une autre idée consiste à mettre TURBO-Forth résident en mémoire vive.

Enfin, dernière remarque: serait-il possible d'avoir sur disquette et pour les membres de l'ASSOCIATION, les programmes parus, car il faut souvent les taper de nouveau et dès que cela dpasse une ou deux pages dans la revue, cela prend beaucoup de temps.

Marc DURANTON - 94470 BOISSY ST LEGER

REPONSE: Premièrement: avoir équipé TURBO-Forth d'un éditeur externe permettait d'alléger FORTH lui-même et de récupérer la place gagnée pour des routines plus utiles. L'éditeur dont est équipé TURBO-Forth reprend les principales commandes de WORDSTAR (taille de EDIT.COM: 56k). Si vous êtes un habitué de TURBO-Pascal, vous ne serez pas perdu. TURBO-Forth ne fait que prolonger la ligne des produits BORLAND, mais n'a aucun lien avec BORLAND, cette société n'ayant jamais envisagé de diffuser TURBO-Forth ou une quelconque version de ce langage. Pour en revenir à l'éditeur, le fait qu'il soit externe et "sélectable" en modifiant une variable alphanumérique, permet, si vous préférez utiliser un autre éditeur, de ne pas perdre vos habitudes. Pour la petite histoire, je programme également en dBASE III et je n'utilise JAMAIS l'éditeur intégré à dBASE pour écrire les programmes.

Deuxièmement: mettre TURBO-Forth en résident est fait. Voir dans les pages précédentes.

Troisièmement: fournir les programmes en disquettes, c'est également fait. Si dans l'avenir les programmes ne seront pas tous disponibles immédiatemment sur support magnétique, du moins essayerons-nous de les transmettre par le serveur SAM*JEDI en même temps que la parution de JEDI. Encore une fois, à l'attention de tous les lecteurs de JEDI, nous sommes très peu à programmer, trop peu même, alors qu'il y a des adhérents qui font certainement des choses très intéressantes. N'hésitez pas à valoriser JEDI par votre participation, même modeste.

L'esprit de l'ASSOCIATION JEDI est de transmettre le plus efficacement possible la connaissance des techniques de programmation dans les langages avancés. Si votre connaissance évolue grâce à JEDI, alors renvoyez l'ascenseur, que notre démarche ne soit pas à sens unique. JEDI est diffusé non par abonnement, mais par cotisation, car vous tous êtes associés, donc tous rédacteurs potentiels.

LE SECRETAIRE

TIR SUR LE PIANISTE:
Jedi, une revue "pro-PC et Compatibles"

A réception du numéro de Janvier. C'est un plaisir toujours renouvelé que celui de recevoir JEDI, et je dois avouer que, cette fois, j'ai savouré avec un plaisir non dissimulé le dessin de couverture et sa légende. On ne sait qui applaudir le plus du dessinateur dont le trait est d'une férocité hors de paire, ou du légendeur et de son astuce. Je crois d'ailleurs avoir reconnu l'auteur!...

Ndlr: pour le reste de la lettre, l'auteur a demandé à ce que sa prose soit censurée. J'ai pris sur moi de conserver ce début élogieux (merci pour le cirage... et pour CABU) et la fin de la lettre:

Voilà ce que m'a inspiré la lecture de la dernière livraison... Et si j'ai pris mon pied à remuer sadiquement mon bâton dans la fourmillière, c'est que j'avais cru y percevoir quelque chose comme un ronronnement, et il fallait que je vérifie si je m'étais trompé ou non!... Des fourmis qui ronronnent!! Encore la Forthe!!!

L'affreux jojo de service
Bernard C. LAMBEY - 34070 MONTPELLIER

REPONSE: si vous avez pris votre pied à remuer sadiquement votre bâton de pélerin dans la fourmilière, elle, par contre, est certainement masochiste.

Si JEDI publie beaucoup concernant les CompPC, c'est parce que la majorité des articles reçus sont destinés à ce système. Malgré mon appel au secours d'il y a six mois, je n'ai pas été submergé d'articles pour APPLE II, SPECTRUM, THOMSON TO7-7/70-8-9-9+, AMSTRAD 464-664-6128-PCW, ATARI (un peu), AMIGA, MacINTOSH, et tout ce qui n'est pas équipé d'un 8086/88/286/386.

Remarquez quand même, que de nombreux articles mentionnent et tête de chapitre le standard, les systèmes de destination et une appréciation d'adaptabilité éventuelle. Regardez l'article sur la virgule flottante écrit par ZUPAN, certaines définitions en code machine sont en commentaires, la définition équivalente figurant en F83.

Enfin, et sur votre demande, et après concertation mutuelle, je retape votre nouvelle prose ci-après;

LE SECRETAIRE

HONTE A CEUX QUI ONT HONTE

Il y a peu de temps, lors d'un entretien téléphonique avec notre secrétaire, au cours duquel trois cent quatre vingt sept idées furent évoquées, je reçus un coup de poing dans l'estomac, sous forme de cet articulet à écrire, sans

aucun échappatoire!... Mais il a tellement raison que je le fais avec ardeur.

Le titre vous a déja, grossièrement, donné le sens de l'admnonestation que j'ai reçu mission de vous adresser. Mais prenons les choses par le petit, le tout petit bout...

* * *

Onésime NIMPORTEKI, adhérent JEDI de longue date, attend sa revue avec une impatience sans cesse fgrandissante, et il a un certain mal à comprendre pourquoi elle n'arrive pas avec une régularité métronomique, et d'une, et aussi pourquoi elle ne fait pas cent pages au lieu de 20-25.

Or il se troyve qu'Onésime voudrait utiliser le Turbo-Forth! De plus il est quelque peu las de se faire botter les fesses par tout son entourage informatique, qui tourne sur PC! Il a donc décidé -tout arrive!...- d'abandonner son petit matériel made in France, génial, si si!... mais tellement original que même ses concepteurs ne peuvent plus l'utiliser dans le contexte actuel!...

Concorde, on peut encore le vendre sous forme de charter, comme le France, mais le TMA-9 à disquette-80k-sans-DOS-identifiable ("réservé à l'Education Nationale qui constitue un marché captif pour des années"!), même les "sous-développés" ne le sont pas assez pour nous le reprendre!...

La mort dans l'âme donc, Onésime va se taïwaniser comme tout le monde, et va enfin se rebrancher sur la communauté informatique. Bonne occasion de se repencher sur le contenu de ses disquettes, car il va falloir les MS-DOSiser. Et il redécouvre des petits trésors qu'il a oublié au fond des pistes, dans des secteurs poussiéreux! Et le voilà, tout joyeux, qui retrouve le plaisir de tel ou tel truc mis au pooint avec passion... et de se demander pourquoi il a cessé de l'utiliser, jusqu'à l'oublier, tel utilitaire fort astucieux qui, lors de sa mise au point, l'avait confirmé dans l'idée qu'un Nimporteki, ce n'est pas un Quiconque!

Bref, si on le lui avait dit, jamais il n'aurait cru qu'il avait accumulé tellement de matériel. Et, de fait, avec les pet

même gonflés jusqu'à être devenus originaux, et aussi les petits trucs perso sur lesquels il est passé, passé et repassé à chaque fois que leur utilisation lui mettait le nez sur un petit point à peaufiner, avec tout cela il a une quantité de matériel étonnante!...

OUI-DA!.. ET IL N'EST PAS LE SEUL DANS CE CAS-LA!...

Il se trouve que des circonstances fortuites l'ont amené à prendre contact avec moi. Et il m'a raconté tout ça! Et quand je lui suggère d'en tirer quelques listings propres et de les envoyer à JEDI, il pouffe de rire, rose d'émotion, la main devant la bouche.
 - Mais, vous n'y songez pas! Ce n'est pas digne d'être publié, il faudrait que je les reprenne pour les optimiser, que j'y retravaille. Et puis ça m'intéresse, moi, mais ça n'intéressera pas les autres!
 - Oui! Vous avez raison! Les autres sont des imbéciles qui...
 - Non, ce n'est pas ce que je veux dire, mais ce n'est pas assez bon.
 - Je vois! Les autres sont des puits de science et vous un minus pour lequel...
 - Mais non! Vous m'énervez, à la fin! Je vous dit que ça ne les intéressera pas!
 - Pourquoi?
 - Mais parceque... parceque...
A mon tour de rire!
 - Mon cher, quand vous aurez terminé votre crise de modestie, quand vous cesserez d'avoir honte de vous même et de ce que votre cervelle est capable de concevoir, vous pourrez peut-être considérer les choses sous l'angle suivant:

1) si tout le monde faisait comme vous, Jedi n'existerait pas!

2) qui sait si une petite contribution, publiée, ne sera pas le facteur déclenchant d'une réalisation grandiose?

3) et si quelque ami me signale telle ou telle amélioration à laquelle je n'ai pas pensé, au total, qui tirera profit de la publication?

4) pour résumer, Lao-Tseu a raison, qui dit: "il y a plus dans deux têtes que dans une!"

34) et Confucius n'a pas tort, non plus, qui énonce ironique: "la gloire ne retient que rarement le nom du soldat qui n'offre jamais sa poitrine à l'ennemi!".

Quant aux vingt-neuf autres raisons de publier vos petits sources que j'ai sautées, elles disaient la même chosde que Confucius, mais moins élégamment!

Et si quelqu'un ose me répondre qu'il a peur de faire des fôtes d'ortaugraphe (Ndlr: ah, bon, je croyais que ça s'écrivait HORTAUGRAFE...), nous serons plusieurs à nous rendre chez lui pour l'étriper dans les règles de l'art! (... à coups de GREVISSE..?)

Avez-vous compris? Ai-je été assez clair? Est-ce vraiment la peine que j'insiste? Non?...

Bon! Alors je persiste et signe: "Honte à ceux qui ont Honte! Ils privent les autres du plaisir de les fréquenter!"

* * *

Ceci étant acquis, comment procéder?

1) Vous avez un compatible? Vous adressez à la revue une disquette de texte ASCII. Il sera remis en forme par la rédaction.

2) Vous êtes sous "Petite merveille made in France"? Téléphonez au secrétaire, il vous dira comment faire en fonction de vos moyens. Mais votre contribution est INDISPENSABLE A LA REVUE, qui est, en fait, une superbe auberge espagnole....

Bernard C. LAMBEY

FORTH

RECOPIE D'ECRAN SUR JUPITER ACE

par B. MOUROT

Matériel: JUPITER ACE (pour nostalgiques...)

Ci-joint un système de recopie d'écran sur JUPITER:

```
0 VARIABLE MEM  800 ALLOT

: RAN
8960 8192
DO
  I C@ MEM I 8192
  - + C!
LOOP
;

: RAP
INVIS CLS 736 0
DO
  MEM I + C@ EMIT
LOOP
;
```

STRUCTURE DES FICHIERS DE DONNEES dBASE

par Marc PETREMANN

J'avais dans l'idée de créer des routines en FORTH permettant d'accéder aux données contenues dans un fichier créé par dBASE III/III+. Mais comme je manque de temps pour les développer, je propose à l'ensemble des adhérents de se pencher sur ce problème et de nous faire parvenir leurs solutions dans les meilleurs délais.

Cette proposition tient lieu de "Challenge". En gros, voici l'ébauche et le fil conducteur du projet. Au départ, je pensais simplement accéder aux données contenues dans un fichier dBASE III/III+(/IV?) pour les traiter. Mais entretemps, Michel ZUPAN a mis au point les routines d'accès direct à un fichier quelconque. Or, sachant que dBASE coûte très cher (vraiment très cher, 9000 Fr environ pour dBASE III+ réseau) et qu'il est incopiable, si on veut exploiter, voire simplement saisir des données sur un second micro, il faut acheter 2 dBASE. Une autre solution consiste à écrire un programme et à le "pseudo-compiler" par DBC et à le faire tourner depuis un DBRUN. Mais même DBRUN n'est pas donné.

Donc, si des adhérents planchaient de concert avec moi (et d'autres), peut-être pourrions-nous mettre au point assez rapidement un "runtime" compatible dBASE écrit en TURBO-Forth. Pour une première approche, il s'agirait de lire et d'écrire dans un fichier dBASE. Pour ce faire, il faut en connaître la structure.

Les enregistrements d'un fichier de données dBASE III PLUS sont précédées d'un en-tête, nommé structure du fichier:

OCTET	CONTENU	EXPLICATION
0	1 octet	Identificateur d'un fichier dBASE III+ 03H sans fichier MEMO (.DBT) 83H avec fichier MEMO (.DBT)
1-3	3 octets	Date de la dernière mise à jour au format AA MM JJ en binaire.
4-7	32 bits	Nombre d'enregistrements dans le fichier de données.
8-9	16 bits	Longueur de la structure d'en-tête.
10-11	16 bits	Longueur de l'enregistrement.
12-31	20 octets	Octets réservés (version 1.00).
32-n	32 octets	Description d'un champ.
n+1	1 octet	0DH comme caractère de fin de définition des champs.

Les champs sont définis comme suit:

0-0	11 octets	Nom du champ en ASCII. Des zéros complètent les emplacements inoccupés.
11	1 octet	Type du champ en ASCII (C, N,L,D ou M).
12-15	32 bits	Adresse des données du champ, adresse positionnée en mémoire.
16	1 octet	Longueur du champ en binaire.
17	1 octet	Nombre de décimales en binaire.
18-31	14 octets	Octets réservés. (version 1.00)

Les enregistrements sont mémorisés séquentiellement à la suite de cet en-tête. Chaque enregistrement est précédé par un octet. Ce dernier contient 20H si l'enregistrement est disponible et 2AH si l'enregistrement est repéré pour effacement. Les données des différents champs sont stockées sans aucun séparateur, ni caractère de fin d'enregistrement. Les différents types de données sont mémorisés en ASCII dans les formats suivants :

Caractère	Caractères ASCII
Numérique	- . 0 1 2 3 4 5 6 7 8 9
Logique	? T t F f et Y y N n pour la version américaine, O o N n pour la version francaise (? est utilisé si le champ n'est pas initialisé)
Memo	10 octets représentant un numéro de bloc du fichier (.DBT)
Date	8 octets au format AAAAMMJJ. Exemple: 19861231

En première ébauche, les caractéristiques syntaxiques du programme souhaité sont les suivantes:

USE <fichier.DBF> ouvre le fichier dBASE <fichier> et analyse sa structure.

La structure en cours d'analyse crée des en-têtes dans le dictionnaire, rattachés au nom du fichier, soit par le HANDLE du fichier concerné, soit par le cfa du nom de fichier mis dans le dictionnaire.

Chaque identificateur de champ sera défini dans le dictionnaire FORTH et lors de son exécution réalisera les fonctions suivantes:

- type C, dépose sur la pile l'adresse et la longueur de la chaîne ASCII pour traitement
- type N, dépose la valeur numérique 32 bits et initialise DPL pour un éventuel traitement de la position du point décimal.
- type L, dépose 0 ou -1 sur la pile.
- type D, dépose une adresse et une longueur de chaîne contenant la date de l'enregistrement concerné. Cette date sera remaniée au format JJ/MM/AA.
- type MEMO non traité.

L'accès à un enregistrement du fichier courant sera exécuté par le mot GOTO. Exemple:

3 GOTO pointe le troisième enregistrement du fichier analysé par USE.

Version simple: on ne traite qu'un fichier à la fois. Par conséquent, la réexécution de USE 'oublie' la structure analysée par la précédente exécution de USE.

Version complexe: on traite plusieurs fichiers. Plusieurs exécutions de USE analysent et hébergent plusieurs structures de fichiers. Prévoir une sécurité pour éviter l'analyse de la même structure deux fois. Prévoir le traitement des cas d'homonymie dans les noms de champs.

Exemple de fichier dBASE:

```
Stucture du fichier ABONNE.DBF
NOM,C,40
ADRESSE,C,40
CODEPOST,C,5
COMMUNE,C,25
NOTEL,C,8
MOISADH,N,2,0
ANNEADH,N,2,0
```

Traitement par FORTH prévue:

```
USE ABONNE.DBF
```

devra analyser le fichier et créer les mots suivants dans le dictionnaire FORTH:

```
NOM ADRESSE CODEPOST COMMUNE NOTEL MOISADH ANNEADH
```

puis 1 GOTO pointe sur le premier enregistrement;

NOM TYPE devra afficher le contenu du champ NOM du premier enregistrement.
MOISADH D. devra afficher le contenu du champ numérique MOISADH du premier enregistrement.

Pour ceux qui connaissent bien dBASE III/III+, nous leur laissons toute liberté pour adapter les commandes dBASE existantes au format FORTH.

■

FORTH

ESSAIS DE GRAPHIQUES SUR CARTE HERCULES

par CLAUDE PILVERDIER

pour: TURBO-Forth
F83 Laxen et Perry
nécessite la présence d'une carte graphique HERCULES

Téléchargement: 3615 SAM*JEDI

NB: Ce programme sera prochainement disponible dans le module de complément M4 de TURBO-Forth.

LISTING: HERCULE.FTH

```
empty only forth forth definitions
\ Adressage long
  ASSEMBLER DEFINITIONS FORTH ALSO DEFINITIONS HEX
b000 constant page0 b800 constant page1
 3b4 constant indx  3b8 constant cntrl
\ Caracteristiques carte HERCULES
  decimal
 \  DEFINITIONS MODE GRAPHIQUE
\ Etablissement des tables de controle
\ du 6845 (controleur d' ecran).
variable gtable 10 allot   variable ttable 10 allot
 : ov!+ over c! 1+ ;  hex
 ttable
  61 ov!+ 50 ov!+ 52 ov!+ 0f ov!+
  19 ov!+ 06 ov!+ 19 ov!+ 19 ov!+
  02 ov!+ 0d ov!+ 0b ov!+ 0c swap c!
 gtable
  35 ov!+ 2d ov!+ 2e ov!+ 07 ov!+
  5b ov!+ 02 ov!+ 57 ov!+ 57 ov!+
  02 ov!+ 03 ov!+ 00 ov!+ 00 swap c!
 : af.table
    0c 0 do dup i + c@ 4 u.r loop drop ;
   3bf constant config  \ port de configuration
 : setmd (s tabl,code,gr/txt --)
 0c 0 do dup i dup indx pc! + c@ indx 1+ pc! loop drop ;

  \ le 6845 est initialise en texte ou graphique
 : gmode 3 config pc! 80 cntrl pc! gtable setmd
 8a cntrl pc! ;                    \ mode graphique
 : tmode  0 config pc! 0 cntrl pc!
 ttable setmd 8 cntrl pc! ;        \ mode texte
 : sono beep key 1b = if tmode abort" commande" then ;
 : so-t sono tmode ;
code toggle bx pop ax pop ax 0 [bx] xor next end-code
\ cette commande permet de voir le graphique et d'inter-
\ rompre l'execution en cours par ESC

\ Routines graphiques en assembleur
 hex      2cf constant xmax 15b constant ymax
 code fill.txt (s page, car -- page )  \ remplit la page
 ax pop bx pop bx push es push bx es mov  \ avec car
4000 # cx mov di di xor cld
 rep ax stos  es pop next end-code
 : g-dark page1 0 8000 0 lfill ; \ vide l' ecran graph.
 variable car.gr  variable all.eff

\ caracteres de remplissage et controle all/effacement
  variable erreur variable erreur1 variable erreur2
 variable xx variable yy
 variable deltax variable deltay
 variable x10 variable x20 variable y10 variable y20
 variable x_1 variable y_1 variable x_2 variable y_2
 variable vdx variable vdy variable pente

\ Calcul de l' adresse du point X,Y en ram
 label calc.adr \ dx=X,di=Y
  dx push di push 3 # di and di di add
 0c # cl mov di cl shl \ (y mod 4 )*2000H
 ax pop 2 # cl mov ax cl shr             \ int(y/4)
 ax ax add                               \ y/4 * 2
 ax dx mov ax ax add dx ax add           \ y/4 * 6
 ax dx mov ax ax add dx ax add           \ y/4 * 18
 ax dx mov ax ax add ax ax add ax dx add   \ int(y/4)*90
 dx di add ax pop 4 # cl mov ax cl shr
 ax di add  ax di add ret \ int(x/16)

 \ retour : di=adr
 label retour cs ax mov ax ds mov ax es mov next end-code
\ le mot ci-dessous allume ou efface
\ le point choisi suivant ALL.EFF
 code pos.point
 (s page,y,x -- page )
 dx pop di pop es pop es push dx push
calc.adr #) call cx pop 0f # cx and 80 # ax mov
ax cl ror all.eff #) dx mov dx dx and
0= if ax es: 0 [di] or else ax not ax es: 0 [di] and then
retour #) jmp end-code

\ Les traits horizontaux sont traites à part
\ car on peut souvent remplir des octets entiers
 label fin-th bp pop si pop bx pop
 retour #) jmp   end-code

 code tr-hor (s page,y,x2,x1-- page)
 dx pop ax pop di pop es pop es push
 \ dx=x1,ax=x2,di=y,ds=page
 bx push si push bp push ax dx cmp
 > if ax dx xchg then       \ trace de gauche a droite
 ax push dx push 4 # cx mov cx push
 ax cl shr cx pop dx cl shr          \ division par 16
 ax bp mov dx bp sub dx pop dx push calc.adr #) call
 di si mov bp si add bp si add bp dec car.gr #) bx mov
 cx pop 0f # cx and
 es push ds pop -1 # ax mov ax cl shr ah al xchg bx ax and
 cx pop 0f # cx and 0f # cx sub cx neg
 -1 # dx mov dx cl shl dl dh xchg bx dx and si di cmp
 0= if dx ax and ax 0 [di] xor fin-th #) jmp then
 ax 0 [di] xor di inc di inc bp dec  bp cx  mov cld  bx ax
mov
 begin cx cx or 0>= while
 ax 0 [di] xor di inc di inc cx dec repeat
  dx 0 [di] xor fin-th #) jmp end-code

\ trait vertical
assembler label all-tv
      begin di push dx push ax push  \ allume
          calc.adr #) call
         ax pop ax es: 0 [di] or
          dx pop di pop di inc
          bx dec bx bx and 0>= while
   repeat
ret         end-code

assembler label etin-tv
      begin di push dx push ax push     \ eteint
          calc.adr #) call
         ax pop ax es: 0 [di] and
          dx pop di pop di inc
          bx dec bx bx and 0>= while
     repeat
ret          end-code

  code tr-vt (s page,x,y2,y1 -- page)
di pop cx pop dx pop es pop es push      \ registres FORTH
bx push si push bp push di cx sub        \ longueur trait
cx bx mov dx cx mov 0f # cx and
80 # ax mov ax cl ror        \ prep. car. graph
dx push all.eff #) dx mov          \ allume ou eff. ?
 dx dx and 0= if dx pop all-tv #) call
   else dx pop ax not etin-tv #) call
   then fin-th #) jmp end-code

\ Trace de droites (algorithme de BRESENHAM
\  BYTE de septembre 1987 )
\ Traitement des coordonnees
\ Récupère des coordonnees en réserve.
 : x_coord x_2 @ x_1 @ - dup deltax ! dup abs / vdx !
  y_2 @ y_1 @ - dup deltay ! dup abs / vdy ! ;
 : xchg_xy x_1 @ y_1 @ x_1 ! y_1 ! x_2 @ y_2 @ x_2 ! y_2 !
 vdx @  vdy @  vdx !  vdy !   deltax  @ deltay  @ deltax !
deltay ! ;

 : nr_c (s x1,y1,x2,y2--- )
  y20 ! x20 ! y10 ! x10 ! ; \ range les coord. fournies
 : rest_c x10 @ y10 @ x20 @ y20 @ ; \ les restitue
 : rec.c y_2 ! x_2 ! y_1 ! x_1 ! ;  \ les prépar à usage
```

```
 : xfer_c rest_c rec.c ;       \ transfert
: prise-c nr_c xfer_c ;
: aff-c cr ." x1=" x_1 ? ."  y1=" y_1  ? ."  x2=" x_2  ? ."
y2=" y_2 ?
   ." dx="  deltax ?  ." dy=" deltay ? ." err=" erreur ? ."
err1=" erreur1 ?
   ." err2=" erreur2 ? cr ;
\ L' algorithme n' est valable que pour pente <= 1
 : ver_coord deltay @ abs deltax @ abs > dup pente !
 if xchg_xy then ;
  defer xfer.pt \ pour utilisation de debug

\ commandes de trace ou d' effacement
 : trace.g all.eff off ['] pos.point is xfer.pt ;
 : sup.g   all.eff on  ['] pos.point is xfer.pt ;

\ segment droit quelconque

 code tr-ob (s page -- page )
ax pop ax push bx push si push bp push ds push
\ sauvegarde registres FORTH
ax es mov x_1 #) dx mov y_1 #) si mov
\ prise des coord. debut
vdy #) bp mov pente #) bx mov
\ prise de param. auxiliaires
deltay #) ax mov ax ax and 0< if ax neg then
 ax ax add ax erreur #) mov ax erreur1 #) mov
\ prepar corrections
 ax erreur2 #) mov deltax #) ax mov ax ax and 0<
if ax neg then
ax deltax #) mov ax ax add ax erreur #) sub
 ax erreur1 #) sub
erreur #) ax mov deltax #) cx mov
\ boucle, tout est pret, on y va
begin ax push cx push dx push si di mov
0 # bx cmp < if dx di xchg then dx cx mov cx push
calc.adr #) call cx pop 0f # cx and 80 # ax mov
ax cl ror all.eff #) dx mov dx dx and
0= if ax es: 0 [di] or
   else ax not ax es: 0 [di] and then
dx pop cx pop ax pop 0 # ax cmp >   \ ***
  if erreur1 #) ax add bp si add
  else erreur2 #) ax add then
vdx #) dx add cx dec cx cx and 0=
 until dx x_1 #) mov si y_1 #) mov
es pop bp pop si pop bx pop next
end-code

\ Selection du type de trait a tracer
: draw (s page,x1,y1,x2,y2 -- page,x2,y2)
 nr_c xfer_c x_coord  deltax @ deltay @ or
 if 0 case
  deltax @ of x_1 @ y_2 @ y_1 @ tr-vt endof
  deltay @ of y_1 @ x_2 @ x_1 @ tr-hor  endof
 endcase
  ver_coord tr-ob then x20 @ y20 @ ;  decimal

\ Essai des differents traits

decimal trace.g

 : fagot page1 g-dark
720  0 do i 00 xmax i - ymax draw 2drop 20 +loop
 13 333 do 00 i xmax ymax i - draw 2drop
 -20 +loop drop ;
\ trace d' un rectangle
: carre-s (s x1,y1,x2,y2 -- )
\ coordonnées d' une diagonale
nr_c xfer_c
 y_1 @ x_2 @ x_1 @ tr-hor y_2 @ x_2 @ x_1 @ tr-hor
 x_1 @ y_2 @ y_1 @ tr-vt x_2 @ y_2 @ y_1 @ tr-vt
  drop ;

variable decb
\ tracé de triangle plein
: esh0 (s page,ydeb -- ) dup >r ymax 2/ swap do
  i dup decb @ +  xmax over -
  2dup > if swap then tr-hor \ else 2drop drop then
   loop r> ymax swap - ymax 2/ do
  i dup decb @ +  xmax over -
  2dup > if swap then tr-hor \ else 2drop drop then
    loop ;
```

```
 \ dessin anime
  : hh (s  -- ) decb off page1 ymax 2/ 0
  do g-dark i esh0 1 decb +!
  stop? if leave then loop drop ;

 \ essais de traits
  hex variable rect 40 allot rect 40 erase decimal
  : rempl.rect 0 do rect i 2* + ! loop ;
  : lit.cote 4 * 2 0
    do dup rect i 2* + + @ swap loop drop ;
  010 100   010 700   250 700   300 650   300 030
  250 100   010 100   200 100   200 700   200 650
  20 rempl.rect
 : polygone trace.g page1  gmode g-dark   0 lit.cote
  10 1 do i lit.cote draw
   loop so-t 2drop drop ;
 \ Remplissage
 base @ hex
 variable adr variable octet
  code adr.point (s page,y,x -- page )
  dx pop di pop ds pop ds push dx push
  calc.adr #) call cx pop
  0f # cx and 8000 # ax mov ax cl shr ah al xchg
  di push ax push retour #) jmp end-code
 base !
 \ dessin avec fleches (pavé numérique)
 variable touche
 : depl-cg
   case \ déplacement du curseur
   48 of -1 all.eff ctoggle  endof
   49 of -1 x_1 +!  1 y_1 +! endof \ graphique
   50 of            1 y_1 +! endof
   51 of  1 x_1 +!  1 y_1 +! endof
   52 of -1 x_1 +!           endof
   53 of 360 x_1 ! 174 y_1 ! endof
   54 of  1 x_1 +!           endof
   55 of -1 x_1 +! -1 y_1 +! endof
   56 of           -1 y_1 +! endof
   57 of  1 x_1 +! -1 y_1 +! endof endcase
   y_1 @ 0 max ymax min
   x_1 @ 0 max xmax min ;
: dessine begin key dup 27 <> while
     depl-cg pos.point repeat drop ;


 car.gr on
 \ cercle et ellipses complets
variable xctr variable yctr variable ray
variable excent 666 excent !
 : exctr dup 0<> if excent @ 1000 */ then ;
 defer x.pt
: (x.pt) (s page,y,x,y1,x1-- page,y1,x1)
      2swap >r >r xfer.pt r> r> ;
 ' (x.pt) is x.pt
: (x.bl) 2swap >r >r dup 15 + swap tr-hor r> r> ;
: cerc.coord (s x,y,r--) ray ! yctr ! xctr ! ;
: cerc.0 (s X,Y-- )
 2dup exctr yctr @ + swap xctr @ +                  x.pt
 2dup exctr negate yctr @ + swap xctr @ +           x.pt
 2dup exctr negate yctr @ + swap negate xctr @ +    x.pt
 2dup exctr yctr @ + swap negate xctr @ +           x.pt ;
: cerc.1 cerc.0 swap cerc.0 swap ;
  \ Cercle complet
: cerc.2 (s --) cerc.coord
 0 ray @ 3 over 2* - erreur ! begin
 2dup < while cerc.1 erreur @ 0>
 if 2dup - 4 * 10 + erreur +! 1-
  else over 4 * 6 + erreur +! then
  swap 1+ swap repeat 2drop ;
: cer.prep swap 2swap rot ;
: rond page1 cer.prep cerc.2  drop ;
 \ Essais cercle
: cercle (s page,x,y,r--page)
  dup >r cerc.coord 3 r> tuck
 2* - >r 0  begin 2dup >= while   \ page,y,x
over  exctr yctr @ + dup >r
 over xctr @ +        x.pt
 r>  over xctr @ swap - dup >r x.pt
over  exctr yctr @ swap -    r>
over >r  x.pt
 dup   xctr @ +  r> swap  x.pt swap
over  exctr yctr @ + dup >r over xctr @ +
 x.pt r>  over xctr @ swap - dup >r x.pt
```

```
over  exctr yctr @ swap -   r> over >r   x.pt
 dup   xctr @ +     r>    swap    x.pt swap
r@ 0> if 2dup    swap    - 4 * 10 + r> + >r swap 1- swap
 else dup  4 * 6 + r> + >r then  1+
repeat r> 2drop drop ;

 \ Essais cercle
 : sqr (s n -- n )
   dup >r 2/ begin r@ over / 2dup - abs 1 > while
     + 2/ repeat nip r> drop ;

defer tr-hz
' tr-hor is tr-hz
: c.blan (s page,xc,yc,r -- page) dup exctr ray ! -rot
 yctr ! xctr ! exctr 0
 do ray @ dup * i dup * - sqr    exctr    vdx !
 yctr @ i + xctr @ vdx @ 2dup + -rot - tr-hz
 yctr @ i - xctr @ vdx @ 2dup + -rot - tr-hz
 loop ;
\ essais divers
decimal variable tbl-sin 182 allot
  tbl-sin 182 erase
: charge-table (s adr, nb -- )
   2*  bounds swap do i ! -2 +loop ;
   0  175  349  523  698  872 1045 1219 1392 1564
1736 1908 2079 2250 2419 2588 2756 2924 3090 3256
3420 3584 3746 3907 4067 4226 4384 4540 4695 4848
5000 5150 5299 5446 5592 5736 5878 6018 6157 6293
6428 6561 6691 6820 6947 7071 7193 7314 7431 7547
7660 7771 7880 7986 8090 8192 8290 8387 8480 8572
8660 8746 8829 8910 8988 9063 9135 9205 9272 9336
9397 9455 9511 9563 9613 9659 9703 9744 9781 9816
9848 9877 9903 9925 9945 9962 9976 9986 9994 9998 10000

tbl-sin 90 charge-table
: lit-table (s adr,n -- n ) 2* + @ ;
: sin (s angle -- n ) 360 mod
 90 /mod tuck 1 and if 90 swap - then
 tbl-sin swap lit-table
 swap 2 3 between if negate then 10000 */ ;
: cos (s angle -- n ) 90 + sin ;
200 ray ! defer trait
: (trait) >r >r >r >r rot r> r> r> r> draw 2drop -rot ;
' (trait) is trait
variable pas
: ellipse (s xc,yc ,af,ad,pas,r,exc--) trace.g
     excent ! ray ! pas ! tuck >r >r dup
     2over 2swap ray @ exctr swap sin rot + -rot
     ray @ swap cos + swap 2swap r> pas @ + r> pas @ +
     do over ray @ i cos + over ray @ exctr i sin +
       2swap >r >r draw r> r>
     pas @ +loop 2drop 2drop ;
decimal
\ gmode g-dark page1 360 174 145 45 10 200 500 ellipse
\ 360 174 370 0 10 150 666 ellipse
\ 360 174 370 0 20 150 333 ellipse so-t drop

: soleil (s x0,y0,r --- ) ray !
180 0 do 2dup swap ray @ i cos
 2dup negate + -rot + rot             \ x0 y0 x1 x2 y0
 ray @ i sin  exctr 2dup
 negate + -rot +                     \ x0 y0 x1 x2 y1 y2
   rot swap trait 10 +loop drop 2drop ;

: soleil4 page1 dup 360 174 200 soleil
      dup 100 174 100 exctr - 100 soleil
      dup 620 174 100 exctr - 100 soleil
      dup 620 174 100 exctr + 100 soleil
          100 174 100 exctr + 100 soleil ;
: soleils 750 100 do page1 0 32767 all.eff @
      lfill i excent ! soleil4 130 +loop ;

eof  \ <-- EOF à supprimer pour chargement de hercula
include hercula
eof
```

Ce programme graphique est encore sommaire, mais il permet de faire des graphiques sur une machine équipée d'une carte Hercules. Il est composé de deux modules: hercule et hercula.

Hercule contient les définitions de base du système graphique:

GMODE fait basculer la machine en mode graphique, car la carte Hercules ne supporte pas simultanément les modes texte et graphique.

TMODE fait l'inverse.

Le mode graphique peut utiliser 2 pages, de 8 Ko chacune, commençant aux adresses B000:0 et B800:0, elles sont désignées par Page1 et Page2; la première est partagée avec le mode texte, ce qui fait que chacun de ces modes écrase la mémoire écran de l'autre. C'est pourquoi seule la page1 est utilisée ici.

G-DARK efface l'écran graphique (comme beaucoup d'autres mots graphiques, il n'a d'effet que sous GMODE, bien que son exécution se fasse sans encombre en mode texte; c'est d'ailleurs un moyen de suivre le déroulement d'un mot graphique avec DEBUG: le lancer sans passer en mode graphique.

POS.POINT allume ou éteint un point en fonction de la valeur de ALL.EFF.

TR-OBL trace un segment de droite quelconque.

TR-HOR trace un segment horizontal, avec un gain de vitesse appréciable, allume et efface alternativement chaque point.

TR-VRT trace un segment vertical.

Ces trois commandes laissent sur la pile, outre le segment de page, les coordonnées des deux derniers points, celà facilite certaines opérations, comme le tracé de polygones.

Aucune de ces fonctions ne contrôle la validité des données qui lui sont fournies, c'est laissé à l'autorité de l'utilisateur. Des dépassements de valeurs, surtout en négatif, peuvent entraîner des plantages variés, mais CTL-ALT-DEL ou RESET sont bien connus des utilsateurs de FORTH et de l'assembleur.

DRAW trace un trait quelconque, en faisant le choix entre les trois précedents.

CERCLX trace un cercle ou une ellipse complet, dont l' excentricité se trouve dans EXCENT, la valeur 666 correspondant sensiblement à un cercle.

ELLIPSE trace un arc de même espèce.

C.BL trace un cercle plein, mais l'excentricité n'en est pas contrôlable, son avantage est une plus grande rapidité.

POLYGONE trace une suite de segments de droite.

ADR.POINT laisse sur la pile l' adresse du point de coordonnées X,Y, à toutes fins utiles.

Le fichier HERCULA.FTH ne contient aucune définition, mais ne sert qu' à faire une petite démonstration de quelques figures graphiques.

LISTING: HERCULA.FTH

```
\ commandes graphiques
dark
10 12 at
.( AU BIP SONORE VOUS POUVEZ ARRETER)
.( LE PROGRAMME PAR <ESC>)
10 14 AT
   .( SINON UNE TOUCHE EN FERA CONTINUER L' EXECUTION. ) SONO
trace.g gmode g-dark
page1 360 174 360 0 10 200 200 ellipse drop sono g-dark
 fagot 666 excent !
360 174 165 rond

trace.g
 page1 360 170 200 cercle
 car.gr on 400 excent !
```

```
 360 170 052 cercle
 360 170 150 c.blan  sup.g
 666 excent !
 360 170 60 c.blan
\ sono tmode .s cr .( fin des cercles) cr sono gmode
 trace.g  decimal
 -1 car.gr !  666 excent !
110 165 225 exctr - 610 165 225 exctr + carre-s
 sono g-dark
soleils
sup.g soleils sono \ g-dark
trace.g soleil4 sono hh
polygone so-t
\ dessin 'a main levée'
dark
10 10 at
 .( Maintenant  vous pouvez faire du dessin à 'main levée'
)
10 12 at
.( les chifres servant de flèches, )
.( 0 bascule tracé/effacement )
10 14 at
 .( 5 ramène le curseur au milieu de l' écran )
10 16 at .( <ESC>  arrêtera l' exercice. ) cr .s sono
trace.g page1 gmode g-dark 360 x_1 ! 174 y_1 !
.s dessine drop so-t dark
```

PROGRAMMER SOUS GEM
EN VOLKSFORTH-83
SUR ATARI ST

par Daniel FLIPO

L'objet de DEMOGEM est de présenter une application sous GEM écrite en VOLKSFORTH-83 pour ATARI ST, avec menu déroulant, boîtes de dialogues et d'alerte, et deux fenêtres : les vrais problèmes de reconstruction d'écran apparaissent dès qu'il y a plusieurs fenêtres, et deux suffisent pour exposer les méthodes à mettre en oeuvre.

I GENERALITES SUR LE GEM

Le GEM se compose de deux parties : l' AES et le VDI.
L'AES gère les menus, fenêtres, boîtes de dialogue, d'alerte, tandis que le VDI gère tout ce qui est dessin. Le VOLKSFORTH-83 (VF83 en abrégé) donne accès à toutes les fonctions GEM, et les noms utilisés sont à peu de choses près ceux du C.

Il est indispensable de disposer d'un ouvrage décrivant les fonctions du GEM (soit [1] soit [2]) et de faire imprimer les fichiers du dossier GEM de VF83 pour faire la correspondance entre les noms en C et les noms VF83.

Il vous faudra également un programme de construction de ressources pour créer les menus, et les boîtes de dialogue, mais pas les fenêtres. Vous avez le choix entre K-RESSOURCES (qui ne fonctionne malheureusement pas avec les nouvelles ROM), RCS 2 de D.R., ou MRCP inclus dans le MEGAMAX C. Les labels des objets créés devront être convertis en VF83, par exemple :
    #define MENU 0   devient en VF83   0 >label :menu
On a l'habitude en VF83 de faire précéder les noms des constantes GEM de : , pour mieux les repérer, mais ce n'est pas une obligation. Les constantes ainsi définies sont regroupées dans le fichier DEMO_H.SCR, non reproduit ici.

Toutes les initialisations fastidieuses du GEM sont assurées par le mot Forth GRINIT qu'il est indispensable d'exécuter au début de toute application GEM. De même on sort (proprement !) du GEM par le mot GREXIT. Après un GREXIT penser à faire un nouveau GRINIT avant d'appeler l'éditeur de VF83 (sous GEM !)...

II COMMENTAIRES SUR LE LISTING DEMOGEM.SCR

II.1 ECRANS 1 à 3 :

On compile les mots nécessaires à DEMOGEM. Les seuls fichiers appelés non inclus dans le VF83 sont :
-SQRT.SCR pour la définition du mot DSQRT (d -- u) qui retourne le plus grand 16 bits non-signé u dont le carré est inférieur ou égal à d (32bits signé ou non). Voir JEDI Nr 27, 32 et 40 pour les définitions.
-CASE_OF.SCR où est définie la structure de contrôle CASE .. OF .. ENDOF .. ENDCASE, également décrite à plusieurs reprises dans JEDI.
-DEMO_H.SCR qui contient les constantes du fichier de ressources.

II.2 ECRAN 4 :

On installe deux tampons de 32ko pour sauvegarder le bureau (tampon Nr 2) et les dessins (tampon Nr 3). Un autre tampon de 32ko (tampon Nr 1) est installé par SUPERGEM.SCR pour sauvegarder l'écran lors de l'affichage de boîtes de dialogue (voir le II.4). Le mot RASTMEMSET initialise ces 3 tampons en fonction de la résolution du moniteur utilisé : DEMOGEM fonctionne aussi bien en haute résolution (N et B) ou en moyenne (couleur). La basse résolution n'est pas prévue. On pourrait ajouter une boîte d'alerte pour empêcher de lancer DEMOGEM en basse résolution ...

II.3 STOCKAGE DES PARAMETRES DES FENETRES

Le fonctionnement des fenêtres nécessite différentes constantes regroupées en vecteurs ou en tableau (cf. écran 6) :

Le vecteur wi_handles contient les 3 index (handle) des fenêtres 0 (bureau) 1 (dessin) et 2 (auxilliaire). Ces index fournis par le GEM à la création des fenêtres lui permettent de les identifier par la suite.

Les vecteurs wi_comps et wi_titles n'ont que 2 composantes (fen. 1 et 2) car la fenêtre 0 n'a aucun attribut (c'est juste un cadre de 1 pixel dont l'utilité sera vue au II.4). Les composantes de Wi_comps définissent les attributs (présence ou non d'une ligne de titre, d'ascenseurs etc, cf [1] p.116 ou [2] p.107) et Wi_titles contient les adresses (16bits) des chaines de caractères formant les titres des 2 fenêtres. Aucune d'elles ne contient de ligne d'information.

Le tableau Wi_sizes (4 lignes 4 colonnes) contient les tailles EN NOMBRE DE CARACTERES de différentes fenêtres :
    -ligne 0 (1er indice = 0): taille maximale de la fenêtre 1 (pour W1_FULL)
    -ligne 1   : taille actuelle de la fenêtre 1
    -ligne 2   : taille (fixe) de la fenêtre 2
    -ligne 3   : taille antérieure de la fenêtre 1 (pour W1_FULL)
Chaque ligne contient 4 nombres de 16bits (le 2ème indice du tableau varie de 0 à 3) : x , y , w , h , dimensions en CARACTERES de l'INTERIEUR des fenêtres, x et y pour les coordonnées du coin supérieur gauche et w et h pour les largeurs et hauteurs.

J'ai fait ce choix bien qu'il complique un peu les choses car on a souvent à inscrire du texte dans les fenêtres et il est alors souhaitable que l'intérieur de la fenêtre borde juste le texte...

Remarquer que lors d'un déplacement ou d'un redimensionnement de la fenêtre active le tampon de message contient la position et la taille du contour EXTERIEUR en PIXELS demandées ! Il faut les convertir en taille intérieure (c'est le rôle de EXT>INT , écran 13) et en nombre de caractères : voir les mots WI_MOVE et WI_SIZE, écrans 13 et 14.

II.4 SAUVEGARDE ET RESTAURATION D'ECRAN

C'est le point le plus complexe de l'AES... Trois cas se présentent :

II.4.1 BOITE D'ALERTE : Pas de problème, le mot ALERT de SUPERGEM.SCR, comme le FORM_ALERT du C se charge de tout, il n'y a rien à faire !

II.4.2 BOITE DE DIALOGUE :

C'est encore très simple, il suffit de recopier la partie de l'écran qui va être cachée par la boîte dans un tampon et de la restituer après : SCR>MEM1 et MEM1>SCR sont fait pour ça !

Le mot DIALOG (écran 20) constitue une macro toute prête pour la gestion complète d'une boîte de dialogue, il utilise SHOW_OBJECT et HIDE_OBJECT définis dans SUPERGEM.SCR qui assurent la restitution d'écran à partir du tampon Nr 1.

II.4.3 CAS DES FENETRES :

Si plusieurs fenêtres sont présentes à l'écran et si on en déplace une, ou si on en ouvre ou en ferme une, ou encore si on lance un accessoire, redessiner l'écran après est une tâche complexe.

A chaque fois qu'une action de l'utilisateur modifie l'aspect d'une au moins des fenêtres ouvertes par le programme, l'AES envoie le message :wm_redraw (&20) et calcule une "liste des rectangles modifiés". Ce message est traité par le mot DISPATCH (écran 22) : le programme exécute alors le mot REDRAW_SCR (écran 12) qui passe en revue cette liste et réaffiche le contenu des fenêtres touchées.

Le mot REDRAW calcule l'intersection du rectangle fourni par l'AES et de la fenêtre courante (celle dont l'index et les coordonnées ont été mis par l'AES dans le tampon message), qu'il ne faut pas confondre avec la fenêtre active, car l'AES passe successivement en revue toutes les fenêtres ouvertes. limite ensuite la zone de dessin à cette intersection (c'est le rôle de SET_CLIP) et lance l'affichage de contenu de la fenêtre courante par WI_FILL défini à l'écran 11.

Quelques remarques :

-Le mot CORRECT (écran 5) a pour seule raison d'être ... une erreur du VF83.
En effet, les rectangles peuvent être définis de 2 façons : soit par les coordonnées (x1, y1, x2, y2) des sommets supérieur gauche et inférieur droit, soit par (x1, y1, w, h) le 2ème sommet étant remplacé par la largeur et la hauteur du rectangle. Pour passer de l'un à l'autre les formules correctes sont :

$x2 = x1 + w - 1$ $y2 = y1 + h - 1$

et VF83 oublie les - 1 !!
(l'origine est (0,0) le point opposé en monochrome est (639,399) les largeurs et hauteurs valant 640 et 400 ...).

-Noter que les mots provoquant des modifications des fenêtres (WI_OPEN, WI_CLOSE, WI_MOVE, etc) n'incluent pas de fonction d'affichage : ils exécutent WIND_SET (dans WI_RENEW) qui demande à l'AES d'envoyer le message :wm_redraw et de préparer la "liste des rectangles modifiés", les modifications d'écrans sont assurées ensuite par REDRAW_SCR.

-Utilité de la fenêtre 0 (bureau) : si le bureau n'est pas déclaré comme fenêtre par WIND_CREATE (cf W0_OPEN écran 11) la mise à jour du bureau n'est pas faite par REDRAW_SCR, car seules les fenêtres DECLAREES comme telles sont prises en compte par l'AES comme fenêtres courantes.
Si la restitution du bureau n'est pas souhaitée, il suffit de supprimer W0_OPEN et W0_CLOSE.

-A la fermeture d'un accessoire l'AES envoie le message :wm_redraw et le programme exécute alors REDRAW_SCR. C'est pourquoi le réaffichage de l'écran est beaucoup plus lent après l'accessoire "Installation d'imprimante" de la disquette ATARI, qu'après celle de la boîte "Informations" de DEMOGEM (qui utilise le procédé II.4.2), surtout lorsque plusieurs fenêtres sont ouvertes...

II.6 LA FONCTION EVNT_MULTI.

Elle attend la réalisation d'un ou plusieurs événements : frappe au clavier, clic ou mouvement de la souris, message, ou "timer".

Pour ne pas avoir à passer par la pile les 16 paramètres qu'elle requiert, le VF83 dispose d'un tableau appelé EVENTS pour stocker ces valeurs. Il faut initialiser ce tableau en début du programme (et à chaque fois que le type d'événement attendu est modifié), c'est le rôle du mot LOAD_EVENTS (écran 19).

Ensuite, juste avant chaque exécution de EVNT_MULTI il faut faire PREPARE pour recopier le contenu de EVENTS dans les tableaux internes du GEM.

EVNT_MULTI retourne sur la pile le type d'événement survenu ; comme il peut se produire plusieurs événements simultanés (en particulier quand le "timer" est actif), on ne testera pas si l'événement survenu est EGAL à :mu_mesag par exemple, mais s'il CONTIENT :mu_mesag, d'où le AND au lieu d'un = dans la boucle BEGIN ... UNTIL de DEMOGEM (écran 23).

II.7 FONCTIONS VDI.

Leur utilisation ne pose guère de problèmes. Vous trouverez dans le listing qui suit comment :

-dessiner un segment (PLINE) ou un cercle (ARC),
-remplir une surface rectangulaire avec un motif (W2_FILL).
-limiter l'aire de dessin (pour que ceux-ci ne débordent pas de la fenêtre (SET_CLIP)

Noter l'utilisation du mode XOR (mode 3 de la fonction SWR_MODE) pour effacer les tracés périmés au fur et à mesure du déplacement de la souris : dans ce mode chaque pixel du nouveau dessin est inversé (noir -> blanc et inversement) par rapport à son état antérieur, donc en traçant deux fois la même fiigure... on l'efface !

Dernière recommandation :

NE PAS OUBLIER DE CACHER LA SOURIS (HIDE_C) AVANT TOUTE MODIFICATION DE L'ECRAN : dessin, affichage d'un objet AES (boîte de dialogue ou d'alerte etc.).
Oubliez le pour voir ... C'est sans risque mais le résultat est moche !

BIBLIOGRAPHIE :

[1] Clés pour ATARI ST Tome 2 Editions du P.S.I.
[2] Le livre du GEM Micro Applications.

0

```
\\                                                   DF 28 03 88
                   EXEMPLE  DE  GESTION
       D'UN MENU, DE FENETRES, D'UNE BOITE DE DIALOGUE.

  Le mot DEMOGEM permet d'ouvrir deux fenêtres, de les déplacer,
de redimensionner l'une d'elle (fenêtre de dessin), toutes les
restaurations d'écrans étant assurées.

  Dans l'une, on peut dessiner des cercles ou des segments de
droites (clic gauche pour commencer, droit pour fixer le dessin)
le choix étant fait par le menu, l'autre est sans fonction (elle
est simplement hachurée).

  Une boîte de dialogue et trois boîtes d'alerte sont gérées :
voir le mot ALERT dans le fichier SUPERGEM.SCR et le mot DIALOG
à l'écran 20 du présent fichier.
```

1

```
  Onlyforth          \ Ecran de chargement            DF 14 03 88
\needs >absaddr      : >absaddr 0 forthstart d+ ;
\needs >label        2 loadfrom assemble.scr
\needs dsqrt         2 loadfrom sqrt.scr
\needs case          include case_of.scr
\needs #def          include gem\gemdefs.scr
\needs gem           include gem\basics.scr
  Onlyforth gem also definitions
\needs :menu         include demo_h.scr
\needs q_mouse : q_mouse &124 0 0 VDI ptsout 2@ swap intout @ ;
\needs pline         : pline >r ptsin r@ 2+ array! 6 r> 0 VDI ;
\needs arc           5 loadfrom gem\vdi.scr
\needs sf_style      3 load
\needs evnt_multi    2 load
\needs tree!         include gem\supergem.scr
  cr  4 &23 thru
```

2

```
\ Fonctions AES nécessaires                           DF 24 02 88
  2 loadfrom gem\AES.scr        \ evnt_multi et consorts ...
  8 loadfrom gem\AES.scr        \ menu words
&17 loadfrom gem\AES.scr        \ form words
&28 loadfrom gem\AES.scr        \ window words
&34 loadfrom gem\AES.scr        \ RSRC words

: objc_draw intin 6 array! &42 6 1 objc_tree 2@ addrin 2! 1 AES
  0= abort" Object_error " ;

: ?err                  0= abort" Graphic_error " ;
: graf_growbox          intin 8 array! &73 8 1 0 AES  ?err ;
: graf_shrinkbox        intin 8 array! &74 8 1 0 AES  ?err ;

  2variable mofaddr     0. mofaddr 2!
: graf_mouse intin ! mofaddr 2@ addrin 2! &78 1 1 1 AES ?err ;
```

3

```
\ Fonctions VDI nécessaires                           DF 13 03 88

: swr_mode    ( mode -- )          intin !    &32 0 1 VDI  ;

: sf_interior    ( style  -- )     intin !    &23 0 1 VDI  ;
: sf_style    ( styleindex -- )    intin !    &24 0 1 VDI  ;
: sf_color       ( color  -- )     intin !    &25 0 1 VDI  ;
: sf_perimeter   ( pervis -- )     intin !   &104 0 1 VDI  ;


\\ Pour modifier la couleur ou l'épaisseur des lignes ...
: sl_color    ( color -- )         intin !    &17 0 1 VDI  ;
: sl_width    ( width -- )         ptsin !    &16 1 0 VDI  ;
```

4

```
\ Installation des tampons, choix de la résolution.  DF 25 03 88
Create memMFDB2  7 , 0 , &640 , &400 , &40 , 0 , 1 ,
                 0 , 0 , 0 ,
 memMFDB2 scr>mem scr>mem2  ( Xtop Ytop Width-1 Height-1 -- )
 memMFDB2 mem>scr mem2>scr  ( Xtop Ytop Width-1 Height-1 -- )
Create memMFDB3  7 , 0 , &640 , &400 , &40 , 0 , 1 ,
                 0 , 0 , 0 ,
 memMFDB3 scr>mem scr>mem3  ( Xtop Ytop Width-1 Height-1 -- )
 memMFDB3 mem>scr mem3>scr  ( Xtop Ytop Width-1 Height-1 -- )

: q_extend     ( flag -- ) intin ! &102 0 1 VDI ;
: setMFDB      ( adr -- )
  >r 0 q_extend intout 2@ r@ 4+ 2! intout @ $10 / r@ 6 + !
     1 q_extend intout 8 + @ r> &12 + ! ;
: rastmemset   ( -- )  \ initialisation des tampons (résolution)
  memMFDB1 setMFDB  memMFDB2 setMFDB  memMFDB3 setMFDB  ;
```

5

```
\ Sauvegarde et restauration d'écran                  DF 26 03 88
: correct ( Width Height --- Width-1 Height-1 )  1- >r 1- r>  ;

: set_clip ( x y w h --- )  \ limite la zone de dessin
  correct  2over  d+  1 s_clip ;

: scr_size ( -- 0 0 lm hm ) \ taille de l'écran entier
  0  0  &80 cwidth *  &25 cheight * ;
: free_size ( -- x y l h )  \  dim "libres" écran (hors menu).
  0 7 wind_get  intout 2+ 4@ ;

: save_desk        scr_size  correct  scr>mem2 ;
: restore_desk     scr_size  correct  mem2>scr ;

\ Coordonnées en pixels du coin haut gauche du champ de vue de
  variable x_top     variable y_top                \ la fenêtre1
```

6

```
\ Composantes et tailles des fenêtres                DF 28 03 88
  Create wi_comps    4 allot  \ composantes des 2 fenêtres
  Create wi_titles   4 allot  \ adresses des 2 chaînes de titre
  Create wi_handles  6 allot  \ "handles" des 2 fenêtres +"desk"
: wi# ( handle -- n )  wi_handles 2-  \ n = N° fenêtre active
  BEGIN 2+ dup @ 2 pick = UNTIL nip  wi_handles -  2/ ;

: array ( dim1 dim2 -- ) \ Crée la structure"tableau de nombres"
  Create  2dup swap , ,  * 2* allot
  Does>  >r
  2dup  r@ 2+ @ <  swap r@ @ <  and  \ test validité des indices
  IF r@ 2+ @  rot  *  +  2*  r> 4+ +
  ELSE r> drop 2drop  ." Erreur d'indices ! "  THEN  ;

  4 4 array  wi_sizes \ i j wi_sizes retourne l'adresse du terme
\ (i,j) du tableau si i<4 ET j<4, et un message d'erreur sinon.
```

7

```
\ Composantes et tailles des fenêtres               DF 23 03 88
: int_size ( n -- x y larg haut ) \ dim.int.fenêtre N°n (pixels)
  0 wi_sizes 4@    cheight * >r  cwidth * >r
                   cheight * >r  cwidth *    r> r> r>  ;

: ext_size ( n -- x y larg haut ) \ dim.ext.fenêtre N°n (pixels)
  >r   0   r@ 1- 2* wi_comps + @   r> int_size
  wind_calc intout 2+ 4@ ;

: w1_fill ( -- )    \ remplissage intérieur fenêtre1
  memMFDB3 >absaddr swap  scrMFDB >absaddr swap  contrl &14 + 4!
  x_top @ y_top @  1 int_size correct 2swap   3 copyopaque ;

: w2_fill  ( -- )  \ remplissage intérieur fenêtre 2
  1 swr_mode  1 sf_color  3 sf_interior  &26 sf_style
  0 sf_perimeter  2 int_size correct 2over d+  bar  ;
```

8

```
\ Taille et position des ascenseurs (fenêtre1)      DF 28 03 88

: h_slize  ( -- )  wi_handles 2+ @  :wf_hslize
  &1000  1 2 wi_sizes @  0 2 wi_sizes @  */  0 0 0  wind_set ;
: h_slide  ( -- )  wi_handles 2+ @  :wf_hslide  x_top @ cwidth /
  0 0 wi_sizes @ -  &1000  0 2 wi_sizes @  1 2 wi_sizes @  -
  ?dup  IF */ 0 0 0  wind_set  ELSE 2drop 2drop THEN ;

: v_slize  ( -- )  wi_handles 2+ @  :wf_vslize
  &1000  1 3 wi_sizes @  0 3 wi_sizes @  */  0 0 0  wind_set ;
: v_slide  ( -- ) wi_handles 2+ @  :wf_vslide  y_top @ cheight /
  0 1 wi_sizes @ -  &1000  0 3 wi_sizes @  1 3 wi_sizes @  -
  ?dup  IF */ 0 0 0  wind_set  ELSE 2drop 2drop THEN ;

: set_slides ( -- )  \ réglage global des 2 ascenseurs
  h_slize  h_slide  v_slize  v_slide  ;
```

9

```
\ Initialisations pour les fenêtres                  DF 28 03 88
  &77 constant x_max          &23 constant y_max
: wi_init ( -- ) \ Initialisation tailles titres et composantes
  1 3 &76 &20  0 0 wi_sizes 4!     1 3 &76 &20  1 0 wi_sizes 4!
  &10 5 &30 8  2 0 wi_sizes 4!     1 3 &76 &20  3 0 wi_sizes 4!
  0" Fenêtre de dessin "    wi_titles !          \ fenêtre 1
  0" Fenêtre auxilliaire "  wi_titles 2+ !       \ fenêtre 2
  :name  :close +  :full +  :size + :move +
  :hslide +  :vslide +       wi_comps !          \ fenêtre 1
  :name  :close +  :move +   wi_comps 2+ !       \ fenêtre 2
  wi_handles 6 erase  ;                          \ handles à 0

: wi_string ( handle function# 0string -- )
  >absaddr swap  0 0  wind_set ;
: set_title ( n -- ) 2*  dup >r  wi_handles + @
  :wf_name  r> 2- wi_titles + @   wi_string ;
```

10

```
\ Ouverture et fermeture des fenêtres                DF 28 03 88

: growbox    ( n -- ) >r little 4@  r> int_size  graf_growbox ;

: shrinkbox ( n -- ) >r little 4@  r> int_size  graf_shrinkbox ;

: wi_open    ( n -- )  \ crée et ouvre la fenêtre N° n
  dup  2* wi_handles + @ IF drop exit THEN
  dup  >r 1- 2* wi_comps + @  r@ ext_size  wind_create
  dup  wi_handles r@ 2* +  !
  r@ growbox  r@ set_title  r> ext_size  wind_open ;

: wi_close  ( n -- )   \ ferme et détruit la fenêtre N° n
  dup  2* wi_handles +  dup @  swap off  dup  wind_close
  wind_delete  shrinkbox   ;
```

11

```
\ Ouverture et fermeture de la fenêtre "bureau"      DF 28 03 88
: w0_open  ( -- )
  0 free_size wind_create dup wi_handles !
  free_size  wind_open ;

: w0_close ( -- )  wi_handles @ dup  wind_close  wind_delete ;

\ Mots utilitaires pour déplacements de fenêtres
: getmessage ( n --- contenu message )   2* message + @ ;

: wi_fill  ( -- )  \ remplit l'intérieur de la fenêtre courante
  3 getmessage wi# CASE 0 OF restore_desk      ENDOF
                        1 OF w1_fill set_slides ENDOF
                        2 OF w2_fill            ENDOF ENDCASE ;
: wi_renew ( -- )  \ met à jour la fenêtre courante
  3 getmessage  5  over wi# ext_size  wind_set  ;
```

12

```
\ Mise à jour de l'écran par les rectangles...        DF 28 03 88

: rect_update ( function# -- x y w h )
  3 getmessage  swap  wind_get  intout 2+  4@  ;

: redraw ( x y w h -- ) \ traitement d'un rectangle
  swap 3 pick +  4 getmessage 6 getmessage +  min
  swap 2 pick +  5 getmessage 7 getmessage +  min
  3 roll 4 getmessage  max   3 roll 5 getmessage  max
  2swap  2over d-  2dup  0> swap 0> and
  IF set_clip  wi_fill ELSE 2drop 2drop THEN ;

: redraw_scr  ( -- ) \ on redessine tous les rectangles modifiés
  :wf_firstxywh rect_update
  BEGIN 2dup or WHILE redraw  :wf_nextxywh rect_update REPEAT
  2drop 2drop ;
```

13

```
\ Changement de fenêtre, déplacement d'une fenêtre   DF 28 03 88
: wi_top   ( -- )   \ changement de fenêtre active
  3 getmessage wi#  \ exclure la fenêtre 0 (bureau)
  IF 3 getmessage  &10  0 0 0 0  wind_set THEN ;

: ext>int ( -- x y l h ) \ conversion coord ext.-->coord int.
  1  3 getmessage wi# 1- 2* wi_comps + @  message 8 + 4@
  wind_calc  intout 2+  4@ ;

: wi_move  ( -- )   \ déplacement fenêtre courante (barre titre)
  3 getmessage wi# >r   ext>int
  rot  cheight /  0 1 wi_sizes @  max
  y_max rot  cheight / -  min  r@ 1 wi_sizes !
  swap cwidth  /  0 0 wi_sizes @  max
  x_max rot  cwidth /  -  min  r> 0 wi_sizes !
  wi_renew  ;
```

14

```
\ Modifications de taille fenêtre1                   DF 27 03 88

: wi_size  ( -- ) \ modification taille (coin inférieur droit)
  ext>int      cheight /  5 max
  rot  cheight / y_max - negate  min  1 3 wi_sizes !
  cwidth /  8 max  x_max  rot cwidth /  -  min  1 2 wi_sizes !
  wi_renew  set_slides ;

: wi_full ( -- )  \ passage à la taille max et vice-versa
  1 2 wi_sizes @  0 2 wi_sizes @  <   \ (coin supérieur droit)
  1 3 wi_sizes @  0 3 wi_sizes @  <  or
  IF 1 0 wi_sizes 4@  3 0 wi_sizes 4!
     0 0 wi_sizes 4@  1 0 wi_sizes 4!
      1 int_size 2drop  y_top !  x_top !  wi_renew
  ELSE 3 0 wi_sizes 4@  1 0 wi_sizes 4!  wi_renew set_slides
  THEN  ;
```

15

```
\ Effet d'un clic dans une glissière d'ascenseur     DF 12 03 88
\ (fenêtre1)

: w1_arrow ( -- )  4 getmessage  CASE
  0 (↑) OF y_top @  cheight /  1 3 wi_sizes @ - 0 1 wi_sizes @
          max cheight * y_top !       v_slide  w1_fill ENDOF
  1 (↓) OF y_top @  cheight /  1 3 wi_sizes @ +   y_max 1 3
   wi_sizes @ -  min  cheight *  y_top ! v_slide  w1_fill ENDOF
  4 (←) OF x_top @  cwidth /  1 2 wi_sizes @ -  0 0 wi_sizes @
          max  cwidth * x_top !       h_slide  w1_fill ENDOF
  5 (→) OF x_top @  cwidth /  1 2 wi_sizes @  +   x_max 1 2
   wi_sizes @ -  min  cwidth *  x_top !  h_slide  w1_fill ENDOF
                  ENDCASE  ;
```

16

```
\ Effet d'un déplacement des ascenseurs (fenêtre1)   DF 27 03 88

: w1_vslide ( -- )  \ effet d'un déplacement glissière verticale
  4 getmessage  0 3 wi_sizes @  1 3 wi_sizes @ -  &1000  */
  0 1 wi_sizes @ +  cheight *  y_top !  v_slide  w1_fill  ;

: w1_hslide ( -- )  \ effet d'un déplacement glissière horizont.
  4 getmessage  0 2 wi_sizes @  1 2 wi_sizes @  -  &1000  */
  0 0 wi_sizes @ +  cwidth *   x_top !  h_slide  w1_fill  ;


\ Variables pour les dessins dans la fenêtre1 :

  DEFER motif
  variable end   variable stop
  variable X0    variable Y0    variable X1    variable Y1
```

17

```
\ Dessins dans la fenêtre 1                          DF 09 03 88
: dess_ligne ( x y --- )  \ segment droite de (X0,Y0) à (x,y)
  X0 @ Y0 @ 2swap hide_c 2 pline show_c ;
: dess_cercle ( x y --- ) \ cercle centre(X0,Y0)passant par(x,y)
  0 3600 2swap  X0 @ Y0 @ 2swap
  Y0 @ - dup m* rot X0 @ - dup m* d+ dsqrt  hide_c arc show_c ;

: dessin  ( --- )  \ dessin d'une droite ou cercle et sauvegarde
  q_mouse drop Y0 ! X0 !
  q_mouse drop 2dup Y1 ! X1 ! motif
     BEGIN    \ attente clic droit pour fixer le tracé
        q_mouse -rot 2dup Y1 @ = swap X1 @ = * 0=
        IF X1 @ Y1 @ motif       \ effacement mode XOR
           2dup Y1 ! X1 ! motif  \ nouveau tracé
        ELSE 2drop   THEN    2 =
     UNTIL  hide_c  1 int_size correct scr>mem3  show_c ;
```

18

```
 DEFER disfig     DEFER rst_menu                         \ DF 28 03 88
: switch  ( -- )   0 getmessage  :mn_selected =
 IF 3 getmessage  dup  rst_menu
    :figures = IF disfig ELSE :cess alert drop  show_c THEN
 ELSE :cess alert drop  show_c THEN ;

: dessins ( -- ) 1 int_size cheight / 0 3 wi_sizes @ - >r cwidth
 / 0 2 wi_sizes @ - >r 2drop r> r> or IF :dess alert drop exit
 THEN  show_c  3 swr_mode  1 int_size set_clip   stop off
 BEGIN     prepare evnt_multi %010010 and  CASE \ filtrer TIMER
   :mu_mesag  OF switch ENDOF
   :mu_button OF dessin ENDOF ENDCASE stop @
 UNTIL ;
\\ 3 swr_mode fait passer en mode XOR, ce qui permet l'efface-
ment par un 2ème tracé superposé au 1er. Sans celà, on obtient
de très jolis faisceaux de droites ou de cercles ...
```

19

```
\  Initialisation du tableau EVENTS.                   DF 23 03 88
: load_events ( --- )
 %00110010        \ attente de BOUTON ou MESSAGE ou TIMER
 1 %11 1          \ pas de double-clic, 2 boutons pris en compte
 0 0 0 0 0   0 0 0 0 0  \ rectangles non spécifiés
 0 0              \ délai 0 millisecondes
 events &16 array! ;

\ Mots pour le menu
: reset_menu ( title# -- ) \ remet le titre en affichage normal
 :menu tree!  1  menu_tnormal  ;

' reset_menu  Is  rst_menu

: check ( f1 f2 -- )  :menu tree! \ placer/oter √ dans le menu
 :ligne swap menu_icheck  :cercle swap menu_icheck  ;
```

20

```
\ Gestion d'une boîte de dialogue                      DF 28 03 88
\ box# = N°(>label) de la boîte, field# = N° du premier champ
\ de texte éditable ou -1 s'il n'y en a pas.

: dialog   ( field# box# -- button )
 free_size set_clip          \ redessin autorisé sur tout l'écran
 tree!  ( curoff)  hide_c  show_object
 form_do dup  deselect
 hide_object  ( curon) ;

\\
 Les mots SHOW_OBJECT et HIDE_OBJECT sont des macros définies
dans SUPERGEM.SCR. La sauvegarde de la partie d'écran recouverte
par l'objet (ici la boîte de dialogue) est assurée par
SHOW_OBJECT dans le tampon d'écran n°1 (installé par SUPERGEM)
HIDE_OBJECT restitue la partie d'écran cachée par l'objet.
```

21

```
: disbureau  ( --- )  4 getmessage  :infos =           \ DF 28 03 88
     IF -1 :box1 dialog  drop THEN  ;
: disfichier ( --- )  4 getmessage  CASE
     :ouvr1 OF 1 wi_open                      ENDOF
     :ouvr2 OF 2 wi_open                      ENDOF
     :fin   OF :bye alert 1 = IF end on THEN ENDOF   ENDCASE ;
: disfigures ( --- )  &200. evnt_timer \ pour que le clic sur le
     4 getmessage CASE          \ menu ne compte pas dans dessins
     :cercle  OF ['] dess_cercle Is motif
                             1 0 check       dessins  ENDOF
     :ligne   OF ['] dess_ligne  Is motif
                             0 1 check       dessins  ENDOF
     :stop    OF stop on     0 0 check                ENDOF
                     ENDCASE  ;

' disfigures Is disfig
```

22

```
\ Traitement des messages                              DF 28 03 88
: dispatch ( -- )
 hide_c  0 getmessage :mn_selected =
 IF 3 getmessage dup  reset_menu
    CASE :bureau  OF disbureau  ENDOF
         :fichier OF disfichier ENDOF
         :figures OF disfigures ENDOF ENDCASE
 ELSE 1 wind_update  0 getmessage  CASE
 :wm_redraw   OF redraw_scr                                ENDOF
 :wm_topped   OF wi_top    ENDOF  :wm_moved OF wi_move     ENDOF
 :wm_closed   OF 3 getmessage wi# wi_close                 ENDOF
 :wm_fulled   OF wi_full   ENDOF  :wm_sized OF wi_size     ENDOF
 :wm_hslid    OF wi_hslide ENDOF  :wm_vslid OF wi_vslide   ENDOF
 :wm_arrowed  OF wi_arrow                                  ENDOF
     ENDCASE  0 wind_update
 THEN  show_c  ;
```

23

```
: settings  ( -- )                                     \ DF 28 03 88
 rsrc_load" demogem.rsc"   \ charg. fichier ressources
 load_events     \ init. tableau EVENTS pour evnt_multi
 0 graf_mouse  1 swr_mode  ; \ souris = flêche, mode normal

: demogem   ( -- )
 grinit settings  rastmemset  wi_init  save_desk
 at? page at  scr_size correct scr>mem3 \ effacer tampon dessin
 :menu tree!  1 menu_bar               \ afficher barre menu
 w0_open      1 wi_open     \ ouvrir fenêtres : bureau + fen.1
 1 int_size 2drop  y_top !  x_top !   end off  show_c
 BEGIN prepare evnt_multi
     :mu_mesag and  IF dispatch THEN  end @
 UNTIL hide_c  :menu tree! 0 menu_bar  scr_size set_clip
 3 1  DO I 2* wi_handles + @  IF I wi_close THEN  LOOP
 w0_close  restore_desk  rsrc_free  grexit  ;
```

objectif Division et racine carrée sur 32 bits.(Suite)
langage Assembleur 6502 (principalement pour ATMOS)
Date 3/1/88 avec tous mes voeux

Division et Racine carree en LM

Voici donc la division 32 bits/32 bits et la racine carrée 32 bits en assembleur 6502 qui suit exactement l'algorithme déjà vu dans un article précédent.Pas de difficulté particulière pour qui sait faire des divisions en assembleur:sauf qu'ici il faut faire les décalages,les comparaisons sur 4 octets. Il y avait quand même une difficulté car sur la pile, ces 4 octets ne sont pas consécutifs,donc pas question de faire de boucle;d'où la lourdeur de certains passages:

Syntaxe : DIV (ud dividende , ud diviseur --- ud quotient, ud reste)
SQRT (ud nombre --- u racine carrée)

Exemple : 1234567. 3456. DIV D. D. donne 775 et 357
1234567. SQRT U. donne 1111

Il est à noter que SQRT peut donner le reste (différence entre le nombre de départ et le carré du résultat) si l'on change la ligne (8# LDY, ...) par celle mise entre parenthèses (4# LDY, ...).

Rien de particulier dans l'assembleur 6502 si ce n'est que j'utilise la zone de travail A5 à A9 que l'on désigne généralement par la lettre 'N'; N désigne A6, N-1 désigne A5, N+1 donne A7 etc...
Les écrans 3 et 4 ne contiennnent que le code assemblé correspondant.Il peut donc être compilé directement sans assembleur sur ATMOS.

Amicalement Jean-Luc SIRET

```
FILE: DIVLM2.FTH / SCRN# 1
\ Division 32 bits,32bits --- 32reste,32quotient
ONLY FORTH DEFINITIONS ALSO ASSEMBLER HEX
LABEL (D) 6 ,X ASL, 7 ,X ROL, 4 ,X ROL, 5 ,X ROL,
          2 ,X ROL, 3 ,X ROL, 0 ,X ROL, 1 ,X ROL, A5 ROL, RTS,
LABEL (C) SEC, 2 ,X LDA, A6 SBC, PHA, 3 ,X LDA, A7 SBC, PHA,
          0 ,X LDA, A8 SBC, PHA, 1 ,X LDA, A9 SBC, PHA,
          A5 LDA, 0 # SBC, CS IF,
          PLA, 1 ,X STA, PLA, 0 ,X STA, PLA, 3 ,X STA, PLA,
          2 ,X STA, SEC, CS NOT IF, SWAP
          THEN, PLA, PLA, PLA, PLA, CLC, THEN, RTS,
CODE DIV 2 ,X LDA, A6 STA, 2 ,X STY, 3 ,X LDA, A7 STA,
          3 ,X STY, 0 ,X LDA, A8 STA, 0 ,X STY, 1 ,X LDA,
          A9 STA, 1 ,X STY, A5 STY,
          20 # LDY, BEGIN, (D) JSR, 6 ,X LSR, (C) JSR, 6 ,X ROL,
          DEY, 0= UNTIL, NEXT JMP, ;C
-->
```

```
FILE: DIVLM2.FTH / SCRN# 3
\ Division
ONLY FORTH DEFINITIONS HEX
HERE DUP DUP 0616 , 0736 , 0436 , 0536 , 0236 , 0336 , 0036 ,
       0136 , A526 , 60 C,

HERE DUP    B538 , E502 , 48A6 , 03B5 , A7E5 , B548 , E500 ,
       48A8 , 01B5 , A9E5 , A548 , E9A5 , 9000 , 680F , 0195 ,
       9568 , 6800 , 0395 , 9568 , 3802 , 05B0 , 6868 , 6868 ,
       6018 ,

CREATE DIV HERE DUP 2- !
       02B5 , A685 , 0294 , 03B5 , A785 , 0394 , 00B5 , A885 ,
       0094 , 01B5 , A985 , 0194 , A584 , 20A0 , 20 C, ROT ,
       0656 , 20 C, , 0636 , D088 , 4CF3 , 0550 ,

-->
```

```
FILE: DIVLM2.FTH / SCRN# 2
\ Racine carree 32 bits
CODE SQRT 4 # LDY, 0 # LDA, BEGIN, DEX, 0 ,X STA, DEY,
          0= UNTIL, A6 STY, A7 STY, A8 STY, A9 STY,
          AA STY, AB STY, A5 STY,
          10 # LDY, BEGIN, (D) JSR, (D) JSR,
          AA LDA, A6 STA, AB LDA,  A7 STA, 0 # LDA, A8 STA,
          A6 ASL, A7 ROL, A8 ROL, SEC, A6 ROL, A7 ROL,
          A8 ROL, (C) JSR, AA ROL, AB ROL,
          DEY, 0= UNTIL,

     ( 4 # LDY, BEGIN, 0 ,X LDA, 4 ,X STA, INX, DEY, 0= UNTIL, )
      8 # LDY, BEGIN, INX, DEY, 0= UNTIL,
      AA LDA, PHA, AB LDA, PUSH JMP,
    ;C

ONLY FORTH DEFINITIONS DECIMAL
```

```
FILE: DIVLM2.FTH / SCRN# 4
\ Racine carree
CREATE SQRT HERE DUP 2- !
       04A0 , 00A9 , 95CA , 8800 , FAD0 , A684 , A784 ,
       A884 , A984 , AA84 , AB84 , A584 , 10A0 ,
       20 C, -ROT , 20 C, , AAA5 , A685 , ABA5 , A785 ,
       00A9 , A885 , A606 , A726 , A826 , 2638 , 26A6 ,
       26A7 , 20A8 , ,       AA26 , AB26 , D088 , A0D7 ,
       E808 , D088 , A5FC , 48AA , ABA5 , 4C C,  0549 ,

ONLY FORTH DEFINITIONS DECIMAL
```